NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SEMANTIC INTEROPERABILITY IN AD HOC WIRELESS NETWORKS

by

Raouf Hafsia

March 2001

Thesis Advisor:	James Bret Michael
Second Reader:	John Osmundson

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2001	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Semantic Interoperability in Ad Hoc Wireless Military Networks			**5. FUNDING NUMBERS**
6. AUTHOR(S) Raouf Hafsia			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT ***(maximum 200 words)***

Ad hoc wireless networks are decentralized networks whose members join and leave the network in an asynchronous manner and for short periods of time. Each node participating in the network acts both as host and a router

Ad hoc networks in theory, support missions of the Armed Forces in situations in which the infrastructure for wire-bound networks is not dependable, it is impractical to build and maintain the infrastructure, or the missions requires that the nodes have a high-degree of mobility.

Ad hoc wireless networks require some level of semantic interoperability so that nodes in the network can "understand" each other. In this thesis we discuss requirements for semantic interoperability in ad hoc wireless networks, and present a case study of how such requirements could be applied. We realized during our study that semantic interoperability components and functions are developed mostly for wired networks, and not taking in consideration the wireless issues such as: processing, power, and networking limitations. In this thesis we discuss wireless user infrastructure, mobile middleware, and wireless application protocols as a solution to realize semantic interoperability in wireless ad hoc networks.

14. SUBJECT TERMS Ad hoc Networks, Routing Protocols, Semantic Interoperability, Wireless Networking			**15. NUMBER OF PAGES** 118
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

SEMANTIC INTEROPERABILITY IN AD HOC WIRELESS NETWORKS

Raouf Hafsia
Captain, Tunisian Army
B.S., Tunisian Military Academy, 1990

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
MARCH 2001

Author: Raouf Hafsia
Raouf Hafsia

Approved by: James Bret Michael
James Bret Michael, Thesis Advisor

John Osmundson
John Osmundson, Second Reader

Dan C Boger
Dan Boger, Chairman
Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Ad hoc wireless networks are decentralized networks whose members join and leave the network in an asynchronous manner and for short periods of time. Each node participating in the network acts both as host and a router

Ad hoc networks, in theory, support missions of the Armed Forces in situations in which the infrastructure for wire-bound networks is not dependable, it is impractical to build and maintain the infrastructure, or the mission requires that the nodes have a high-degree of mobility.

Ad hoc wireless networks require some level of semantic interoperability so that nodes in the network can "understand" each other. In this thesis we discuss requirements for semantic interoperability in ad hoc wireless networks, and present a case study of how such requirements could be applied.

We realized during our study that semantic interoperability components and functions are developed mostly for wired networks, and are not taking into consideration the wireless issues such as: processing, power, and networking limitations. In this thesis we discuss wireless user infrastructure, mobile middleware, and wireless application protocols as a solution to realize semantic interoperability in wireless ad hoc networks.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author would like to thank Prof. James Bret Michael for his guidance and inspiration through every step of this research, Prof. John Osmundson for his help, and most importantly my late parents, Abdelhay and Fatma Hafsia, and of course my wife, Meriem, and daughters, Hela and Fatma, whose patience, encouragement, and love made all the difference.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

With recent performance advancement in computer and wireless communications technologies, mobile wireless computing is expected to become even more widely applied. One of these wireless communication technologies is the wireless ad hoc network, which is a collection of wireless mobile nodes that can dynamically form a temporary network without the use of any existing network infrastructure or centralized administration. However, due to the limited transmission range of wireless network interfaces, multiple network hops may be needed for one node to exchange data with another across the network.

Ad hoc wireless networks can fill the requirements of some applications for mobility and ease of creation and dissolution, such as fire and rescue operations, and tactical communication for military operations. In the case of ad hoc networks used in the battlefield, the tactical and operational information can be passed to decision makers through the battlefield. However, before ad hoc networks can be easily deployed in military and civilian operations, one must evaluate other functional and non-functional system requirements, such as performance, dependability, and interoperability.

An inherent challenge in the evolution of any product category in the networking industry (wired or wireless) is making sure that products work together. The term 'interoperability' means that the nodes in the network can correctly interpret commands and data transmitted among them.

However, wireless networks pose a set of interoperability challenges that do not apply to hard-wired networks. The ever-changing state of the network requires detailed

definitions of interoperability. In this case a lack of interoperability may pose problems, depending on the application for which the product is being used. For instance, for a closed wireless network that will not communicate with another one, interoperability is not an issue. However, for a real networking environment where several possibly heterogeneous networks need to communicate with each other, an interoperable environment is needed.

Interoperability is an important consideration for networks used by the military. Situations can arise in which a nation must integrate its wireless networks into a supra network made up of networks controlled by other nations' armed forces. Since the strategic base is primarily a commercial network, an army needs to ensure that all of its communications will interoperate with international communications standards.

In this thesis we explore the application of formal methods, protocols, and gateways for unambiguously specifying the necessary interfaces to ensure seamless interoperability among all army and joint networks. We present a background of ad hoc networks and semantic interoperability, we give an overview and comparison of wireless ad hoc routing protocols, we discuss requirements of semantic interoperability in ad hoc wireless networks, we discuss a case study describing the implementation of an ad hoc wireless network, and we finish with a conclusion and raise research issues to be addressed.

II. BACKGROUND OF AD HOC NETWORKS AND SEMANTIC INTEROPERABILITY

A. BACKGROUND OF AD HOC NETWORKS

Ad hoc wireless networking is an emerging computing paradigm that supports the rapid on-the-fly creation and dissolution of networks that are intended to have a short-lived existence. The mobile nodes retain their autonomy except for the minimum level of cooperation that is required to pass messages back and forth between nodes.

A mobile ad hoc network can be defined as a wireless network composed of mobile nodes and requires no fixed infrastructure. These nodes are dynamically self-organizing in arbitrary and temporary network topologies. The vision of ad hoc networking is to support dependable and efficient operation in mobile wireless networks by incorporating routing functionalities into mobile nodes. Thus, a network node plays two primary roles simultaneously: as a router, forwarding packet traffic generated by other hops in the network, and as a simple host that is able to communicate with the other hosts in the same sub network. Such networks are envisioned to have dynamic, sometimes rapidly changing, random, multi-hop topologies, which are likely composed of relatively bandwidth-constrained wireless links.

Within the Internet community, routing support for mobile hosts is presently being formulated using "Mobile IP" technology. This is a technology to support nomadic host "roaming", where a roaming host may be connected through various means to the Internet other than its well-known fixed-address domain space. The host may be directly physically connected to the fixed network on a foreign subnet or be connected via a

wireless link, dial-up line, and so on. In order to support this form of host mobility (or nomadicity), the system requires address management and protocol interoperability enhancements. Before ad hoc network technology can be easily deployed, however, improvements must be made in areas such as wireless technology, location and configuration management, addressing and routing, security, and interoperability. The interoperability domain is the main topic of the research reported in this thesis.

Interoperability is an important factor for any network to communicate with others. It represents a universal interaction, and once available, it allows a device or application to adapt its functionality to exploit services it discovers as it moves into a new environment. Interoperability in ad hoc networks can be defined in data communication. Roaming means that clients must have the ability to roam among access point location without loosing connectivity or data integrity, configuration, products, and coexistence. In addition wireless networks must coexist with other wired and wireless products, that is, they must be designed so as not to interfere with one another.

1. Characteristics of Ad Hoc Networks

An ad hoc network consists of mobile platform (e.g., a router with multiple hosts and wireless communications devices) composed by nodes that are free to move, and represent an autonomous system. See Figure 1.

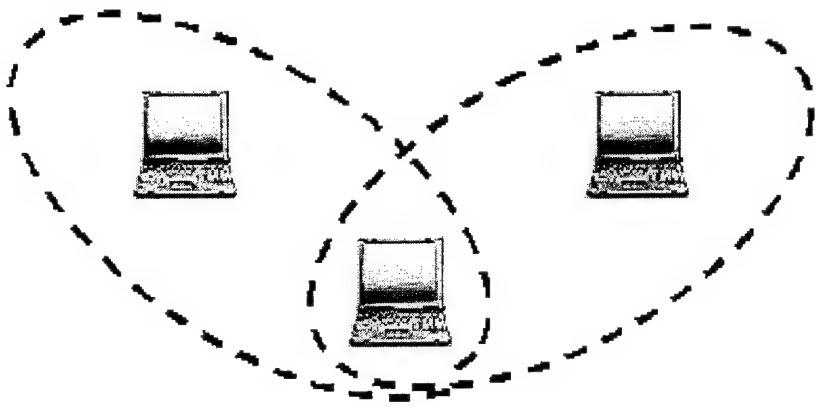

Figure 1. Example of a Simple Ad Hoc Network With Three Participating Nodes.

The system may operate in isolation, or may have a gateway to and interface with a fixed network. Each node in an ad hoc network is equipped with wireless transmitters and receivers using antennas, which may be omni directional (broadcast), highly directional (point-to-point), possibly steerable, or some combination thereof. At a given point in time, depending on the nodes' positions and their transmitter and receiver coverage patterns, transmission power levels and co-channel interference levels, wireless connectivity in the form of a random, multihop graph or "ad hoc" network exists between the nodes. This ad hoc topology may change with time as the nodes move or adjust their transmission and reception parameters.

Generally an ad hoc network has the following characteristics:

a. Dynamic Topology

A dynamic topology implies that nodes are free to move arbitrarily in a specific area according to predefined technical specifications. Thus, the network topology, which is typically multihop, may change in a random manner and rapidly at unpredictable times.

b. Bandwidth Constraints

Wireless links will continue to have significantly lower capacity than their hardwired counterparts. In addition, the throughput of wireless communication, after accounting for the effects of multiple access, fading, noise, and interference conditions, and so on, is often much less than a radio's maximum transmission rate (i.e., the theoretical capacity). The aggregate demand placed on the network resource by applications will likely approach or exceed network bandwidth at some point in time.

c. Energy-Constrained Operations

Some or all of the nodes in an ad hoc network may rely on batteries or other exhaustible means for their energy. Therefore, the most important system design criteria for optimization may be energy conservation.

d. Security

Mobile wireless networks are generally more prone to threats on physical security than are fixed-cable nets, such as eavesdropping, spoofing, and denial-of-service attacks.

These characteristics create a set of underlying assumptions and performance concerns for protocol design and their interoperability with existing protocols.

2. Ad hoc Network Architecture

There are two main types of architecture that can be applied to ad hoc wireless networks: the two-tier hierarchical network architecture and the flat network architecture.

The two-tier hierarchical network architecture achieved by network partitioning/clustering can be used to reduce the control information exchange and the

signaling/control overheads. This architecture can improve critical functions such as media access, routing, mobility management and connection setup.

While all nodes are typically switches/routers, one node in each cluster is assigned as the *clusterhead*, and traffic between nodes of different clusters must always be routed through their respective clusterheads. The number of tiers within the network can vary according to the hierarchical structure of the users, resulting in the hierarchical network architecture as shown in Figure 2.

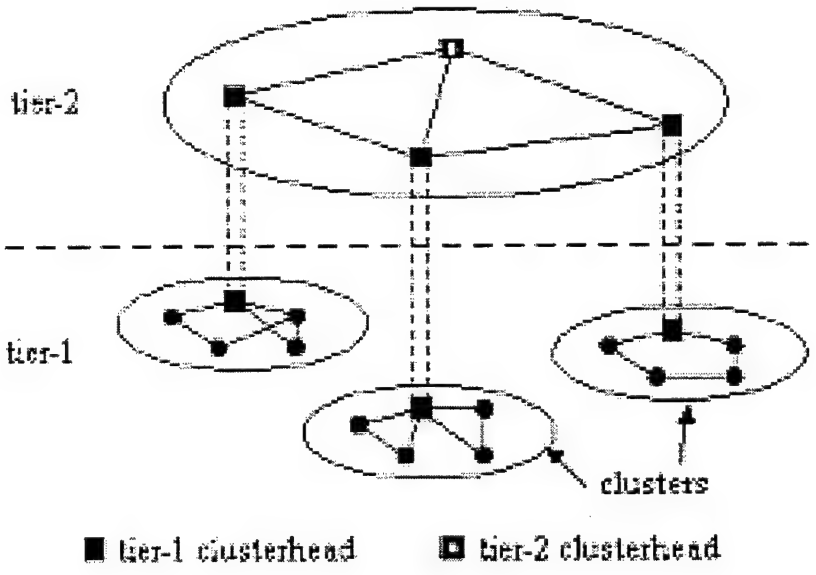

Figure 2. Two-Tier Hierarchical Network.

On the other hand, the flat architecture network makes all nodes equal and connections are setup between nodes that are in close enough proximity to establish radio communications, constrained only by connectivity conditions and security limitations. One advantage of flat networks is the ease in creating multiple paths between communicating nodes, thereby alleviating congestion and providing robustness in the

presence of failures. Route selection can also be made according to the traffic requirements, e.g., low delay and low capacity paths can be used for voice traffic, while voluminous data such as maps can be sent over high capacity but longer delay routes. Figure 3 illustrates flat network architecture.

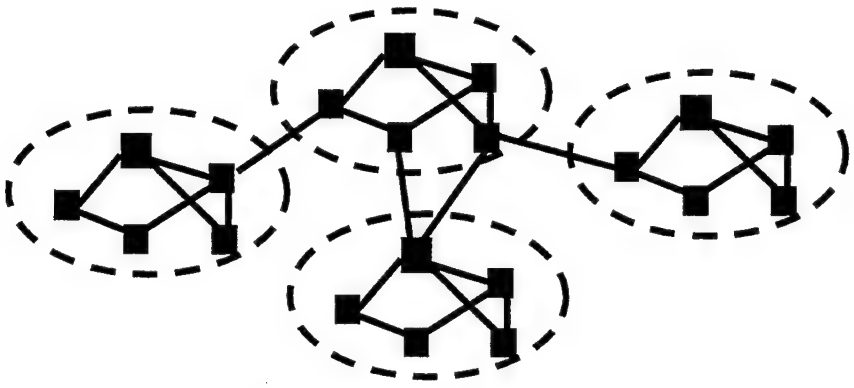

Figure 3. Flat Network Architecture

3. Ad Hoc Network Considerations

The ad hoc wireless networks, also called Mobile Packet Radio Networking, have advantages and disadvantages, when compared with infrastructure and fixed networks.

a. Advantages

- Rapid configuration: Ad hoc networks can be installed quickly in places without previous infrastructure
- Failure tolerance: the failure or the disconnection of a station can easily be bypassed with the dynamic reconfiguration of the network. In a fixed network, in contrast, if a failure in a router occurs, then the traffic redirections is, when possible, a complex operation.
- Connectivity: if two stations are inside of the reach area of the radio waves, then automatically they have a communication channel between them. In a fixed network, even though two stations are next to each other,

it is necessary that the stations have a guided mean to exchange information.

- Mobility: The most important feature in ad hoc networks is the mobility of the different nodes in the network area.

b. Disadvantages

- Broadband: Wireless communication channels normally have less broadband than links in guided ones. In internal environments (indoor), while the speed of a wireless network typically varies from one to two Mbps, in fixed networks this value is in the Gbps range.

- Errors in the wireless link: the range of errors in a wireless link is typically of one wrong bit in each 105 or 106 transmitted bits, while on an optic fiber this range is typically of one in each 1012 to 1015 transmitted bits.

- Routing: in a fixed network the topology hardly changes. The nodes are normally in the same positions in the network. In ad hoc networks, nodes can move freely from one location to another in a non-deterministic manner. If at a given moment node A can be connected to node C via node B, nothing guarantees that this link continues to be present for the duration of A-C communication. Node A, node B, or node C can move and be outside of the area of reach of the others. In this case, it is necessary to find another path from node A to node C, representing an overhead associated with defining the new route.

4. Application of Ad Hoc Networks

The technology of ad hoc networks is somewhat synonymous with Mobile Packet Radio Networking (a term coined during early military research in the 1970s), Mobile Mesh Networking, Mobile, Multihop, and Wireless Networking. There is a need for dynamic ad hoc networking technology. Some applications of ad hoc network technology could include industrial and commercial applications involving cooperative mobile data exchange. These kinds of networks can offer an extremely flexible method for establishing communications for groups that form almost spontaneously to perform a task of limited time duration, such as fire and rescue operations or tactical communications for military operations.

There are likely other applications for ad hoc technology or other scenarios requiring rapidly deployable communication with dependable, efficient, and dynamic networking. The main point here is that ad hoc networks can be deployed in areas in which there is little or no communication infrastructure, or the existing infrastructure is expensive, inconvenient to use, or not dependable.

5. Mobile Computing

> In the mobile communication era, mobility usually refers to the movement of people and the communication terminals. In the computing era, however, advances in technologies have resulted in 'carriable' computers, commonly known as laptops, notebooks, or laptops. [1]

Mobile computing describes computing while on the move. The term computing may refer to performing different activities (e.g., word processing, database retrieval, e-mail, mathematical calculations) with different levels of complexity.

---

[1] C-K. Toh, "Wireless ATM and Ad-Hoc Networks", Kluwer Academic Publishers. 1997, pp. 7-15.

Mobile computing is characterized, however, by the degree of connectivity while on the move. In this context, information may have to be shared among mobile hosts, and interactions are necessary among them.

Mobile computing liberates users from the confines of wired networks. At the same time, transparent access to heterogeneous, distributed information sources has become a reality in wired computing. Obviously, this kind of access that is offering a certain degree of interoperability is highly desirable for mobile users, too.

However, current architectures for transparent information access do not take the specific needs of mobile users into account, while existing architectures for mobile computing do not support seamless* access well. For the future growth of the usage of mobile computing, it is important that a standardized middleware platform extending different types of mobile architectures be set up and defined. This platform has the role to ensure mobile access in heterogeneous environments.

B. BACKGROUND OF SEMANTIC INTEROPERABILITY

1. Definition of Semantic Interoperability

Semantic interoperability within a distributed system is achieved via protocols that provide each component within the system with a means for correctly interpreting data received from any other component. Each system uses standards to accomplish certain tasks and services when it comes to communicating with other networks.

---

*Seamless means transportation of data, whether across one or multiple networks, is transparent to users

For example when a source station intends seventy to mean seventy meters, the receiving station must not interpret the result as seventy feet. A semantic interoperability service is defined by a set of required functional components such as argument describers, conversion functions, and how these functionalities are mapped to the real world.

Semantic interoperability rules should guarantee that if a system sends a simple value (say, the integer 70), then the receiving system would also see a '70', even though the actual (i.e., syntactic) representation of the two values may be completely different, for instance, the decimal value 70 represented in binary on one system and hexadecimal notation on the other system.. Consequently, it is important that the semantics of the data being transferred from one system to another one be preserved.

Semantic interoperability is needed whenever the components (hardware and software) of two or more networks have different expectations about the data they are to exchange.

2. Role of Semantic Interoperability

The following four paragraphs illustrate the role of semantic interoperability.

First, consider an application that needs to display various images that it receives. For each source, it sends a message requesting an image file and some numbers indicating the size of the image. It expects the image to be in GIF format and the size to be in centimeters. However, certain sources may provide PostScript or JPEG images, and send the size information in inches or pixels. Without semantic interoperability, such images are unusable. With semantic interoperability, the images would be converted somehow into the form expected by the application. Also, if the receiver specifies

requirements that cannot be met (e.g., higher precision than is available, or high confidence in authenticity), it should be warned of the inadequacy so as not to produce incorrect results.

Second, consider an application that accesses several databases to receive information about a military mission. Various distances may be received, such as target distance, and cruising range. These numbers may have different formats and interpretations, for instance, great circle or route distance, nautical miles, or kilometers. The differences in units can be resolved by automatically called conversion routines. However, if certain types of distances cannot be converted then users need to be alerted to the semantic mismatch, thereby preventing erroneous interpretations of the data.

Third, error messages need to be understandable to applications, but there is no universal standard for heterogeneous environments. Moreover, it is necessary to permit server-specific return codes. Semantic interoperability requires that each service's error code be mapped into the application-understandable code that best matches it, with warnings for poor matches.

Fourth, consider an application, which interoperates with multiple information retrieval services. Some services may require that a string have a fixed number of characters, left padded with blanks; others may have variable length strings, or pad on the right with zeros. Escape characters and wild cards may differ. The semantic needs to be able to detect and resolve these differences.

There is a common theme among the preceding examples. With current technology, the clients and servers, whether in the same network or in distinct ones, need to be able to communicate via well-defined interfaces. Ideally, each interface should

describe how the application interprets its own data (e.g., target distance). In other words, the abstract interface between systems would be in terms of values that described their own representation and assumptions.

3. Components of Semantic Interoperability

Semantic interoperability as shown above is needed whenever one needs to have two or more systems communicate with each other, even though it is known that each system might have different expectations regarding the data to be passed. Thus, the system must act as a mediator between different systems. The mediation involves determining the expectations that the two systems communicating with each other have for a specific value v, choosing how to convert value v to v' or raising an exception if conversion is not possible, and finally choosing where and when the conversion will occur.

The mediation activities are coupled to the architecture, software applications, computing platform, and other characteristics of the communicating systems. Furthermore the functions, once they are set and well defined, define the level of semantic interoperability.

4. DoD Approach to Interoperability

The primary DOD guidance towards interoperability can be found in the TAFIM*, DII COE*, SHADE*, and joint technical architecture (JTA).

TAFIM* stands for Technical Architecture Framework for Information Management
DII COE* stands for Defense Information Infrastructure Common Operating Environment
SHADE* stands for Shared Data Environment

These guiding strategies promote interoperability through the use of common software, common computer platforms, common communications, and a common set of standards for use on the interfaces.

The COE formulates a portion of the middleware in support of applications executing on the platform: a set of products in support of services such as data management, data interchange, user interfaces, security, and network services. The DII COE and SHADE state that interoperability can be achieved through the use of a common set of products that interface with each other through other common sets of products, or by use of standards-based interfaces across the operating environment set of COTS (commercial-of-the-shelf) or GOTS (government of-the-shelf) products. Further, DII COE also promotes a common architecture for all interactions: the client-server paradigm evolving to a client-broker-server paradigm.

The Defense Science Board (DSB) technical architecture defines a technical framework for the development of systems and for achieving interoperability through standards and common interfaces. Similarly, the TAFIM states that interoperability can be achieved through the development of a common, multipurpose, standards-based technical infrastructure. The TAFIM provides the basis for DoD interoperability of information systems by defining common services, standards, and configuration for the DoD technical infrastructure. TAFIM provides a technical reference model (TRM) for a layered architecture of software services specific to an operating system, and a set of information technology standards applicable to any product realization of these services. This is a necessary but not sufficient condition for interoperability.

The US DoD has as one of its goals to achieve widespread interoperability. The US DoD consists of several domains, such as command and control, logistics, transportation, finance, and procurement. The characteristics of any one of these domains include extremely large quantities of data, thousands of legacy systems, heterogeneity of platforms and operating systems, service-specific processing rules and constraints, and distributed computing. Thus, there is a goal to attain interoperability among system components within a domain, interoperability across domains, and interoperability in data sharing.

Various aspects of distribution transparency are being considered in order to provide infrastructure services to the mission applications wherein the details of the distribution are partially or fully hidden. An example of this latter consideration is the goal to make the location of data transparent to the application accessing the data, which is accomplished through the use of metadata. The US DoD assumes that migration toward open system environments (OSE) remains an ever-present goal, because of the enhancement of competition, interoperability, and portability. The following directive and instruction was published to support the goal:

> DoD Directive (DoDD) 4630.5, *Compatibility and Interoperability of Tactical Command, Control, Communications, and Intelligence Systems*, promulgated in November 1992, requires that procedures be established for the development, coordination, review, and validation of compatibility, interoperability, and integration of Command, Control, Communications, and Intelligence (C3I) systems. It further stipulates that *all* C3I systems developed for use by U.S. forces are considered to be for joint use.[2]

---

[2] Defense Information Systems Agency Center for Standards, "Technical architecture framework for information management. Volume 1." Version 3.0. April 1996.

It may be the case that multiple perspectives and appropriate solutions are needed to achieve interoperability. There is some speculation that an approach based on current research on system architecture and domains will provide the necessary perspective on achieving a high degree of interoperability, especially in a system composed of heterogeneous subsystems.

A system engineer focused on interoperability must consider a number of distributed-system scenarios:

- Interaction between one application and another: When two applications use the same Common Operating Environment (COE), there is no need to consider interoperability across the services of the COE; this is because the target application message interface is specifically known to the source, and the target application is known and can be accessed.
- Interaction between an application and its Common Operating Environment (COE): In order for an application to achieve interoperability with another application, it must interoperate with the services provided by its OE, or COE. This is typically achieved by having the application utilize the application program interface (API) of the service. The intent of the COE is to make it changeable without impacting application, instead of expecting that every change in the COE will necessitate an application change.
- Interaction between a service within a COE and the service or services of a target (different) COE: Addressing interoperability between two applications utilizing different COE work packages becomes more

complex. In this case engineering must address the interface, behavior, naming, service offering, and architecture of each of the different COE work-package services.

- Agreement on the interfaces, in terms of name, syntax, and semantics: In order to achieve interoperability on the interface, both parties must agree on the same standard, including the same mandatory services, syntax of the transfer protocol, semantics of the transfer protocol, and a means in the protocol to identify the standard in use. Although different commercial or government products may be compliant with a standard, interoperability across two different products using the same standard interface does not guarantee interoperability. This is because different standard services or transfer syntaxes could be used. Therefore, in order to ensure interoperability, products compliant with a standard on the interface must also comply with the use of standard services, or an agreement of how to handle differences. There are several documented cases in which different vendor TCP/IP products do not interoperate, despite their strict adherence to the TCP and IP protocol specifications.

- Architecture of the components and their interactions involved in interoperability: When a domain combines two or more different architectures, along with differences in the use of servers, the engineer for the domain must address a number of interoperability considerations, especially issues regarding architectural differences. For example, although application A and application B (from different environments)

can interoperate and exchange documents in a given format (e.g., document type), this communication between A and B might not be possible because of differences in their respective architectures for services.

- Distribution transparency, or if the interoperable transactions should be transparent: Distribution transparency is the ability to hide some aspects of the system components from other components. Transparency minimizes the knowledge needed by a source service about aspects of the interoperable target service. Transparency provides levels of independence for the application. However, transparency is not free: it may require the use of middleware.

Some additional interoperability considerations in the defense community are as follows: vertical interoperability which deals with inter-operation between a service requester and a service provider, horizontal interoperability which deals with peer components, and architectural interoperability which deals with interoperation between two systems which do not exhibit the same peer components in the same style of architecture.

Some of the guidelines for addressing these aspects of interoperability are as follows:

- Use of standard APIs for all Operating Environment Component (OEC) services for a given service

- Common set of standards across the interface, wherein the services of the standard are identified for use
- Standard representations of data and information formats across the interfaces for all interfaces, such as HTML for web-based interfaces
- Standard naming of system components
- Defined standard error handling and recovery
- Architecture that structures the services and their interrelationships
- Use of only those architectures that can be composed

5. Using Software System Architecture to Support Interoperability

The software system architecture represents the application architectures for the tools, components, and connectors. It is decomposable into the technical architecture to define the standards on the interfaces, the details of the infrastructure services, the COE products, hardware platforms, networks, and topologies of both and data and code.

The decomposition of the software system architecture may result in the selection of the client-server style (or three-tiered architecture, peer-to-peer, transaction-based, workflow-based, etc.). Once the style is chosen, the technical architecture determines how the application interacts with the services provided by the distributed system, what interfaces the application uses to interact with the distributed system (i.e., a COE), how the distributed system service components are placed on computer platforms, what networking capabilities are required, and what repository systems are required. All of these specifications are implementation styles, and are used to address the problems defined in the software system architecture.

Client-server, for example, requires a tight coupling between the clients with its server. Client-broker-server requires an interface from a client to a broker, which in essence mediates to different servers. The broker becomes a client to each of the servers. This minimizes the interface development of the client to the set of servers, and enables the client to perceive all the servers as one. The knowledge of the interfaces and mediation are maintained by the broker. The server component remains unchanged, in that the broker is acting in the role of client when it interacts with the server. This architecture provides a high degree of interoperability, and some degree of transparency.

6. Domain and Cross-Domain Interoperability

The technologies used to address the interoperability problem within a domain are currently those of common hardware, common software, composable architectures, and standards on the interfaces. For example, it may be the case that within a given domain, a single Defense Information Infrastructure Common Operating Environment (DII COE) work package is required.

When one introduces multiple domains, the challenge is much greater. It is not reasonable to expect the same workpackage will be utilized across domains. There are fundamental differences in computation requiring different capable products: real-time, large-scale transactions, massive data transactions, near real-time interactive responses, and so on. The interoperating systems must resolve differences in representation; communicating results across distributed systems with different data representations and different database schema representations; functional, performance, evolutionary, and administration differences; and others. Interoperability across domains may require an

enhanced set of technologies over those for interoperability within a domain. It is postulated that these technologies may include:

a. Federation

Federation is about what will be shared with other autonomous domains, and how members will negotiate the possibly short-lived contract to realize that federation. Federation is also about how to integrate autonomous domain members, what to do in the federation, and how to realize system interoperability.

b. Negotiation

Negotiator services are transactions that perform the actions: request for service, agreement to perform the service, delivery of the result, and agreement that the results conform to the request. Negotiators interact with a set of agents, providing the functions to support the negotiation process.

c. Trader

A trader is a component that links client requests with the interface identifier of the service provider. A well-defined protocol is defined to support the trader services: e.g., import, search, select, add, remove, modify, export, and withdraw. A trader provides trading within a domain, for which there is only one trader. The trader takes attribute information about the component and adds the value. The export interface is selected by the trader and provided to the importer for use.

d. Mediation

Mediators perform translations between schema, data formats, and the like. Mediators also reconcile, integrate, and interpret information from multiple, diverse sources.

e. Distributed Object Manager (DOM)

DOM provides the capability for users to identify and access available services in a dynamically configurable distributed environment. It requires the trader and other components to support the establishment between two or more interfaces.

> Many of these technologies are still in research. However, perhaps our approach to interoperability is to focus on domain interoperability. Cross-domain interoperability, short of bilateral agreements between every two domains, may be facilitated when some of these technologies appear as products. Currently, mediator products, trader products, database federator products can be found, although some are just now emerging. We should keep informed of these technologies, and consider them for cross-domain interoperability for future DOD systems.[3]

7. Tactical and Strategic Interoperability

The military community relies on communication to carry out its missions. Communication can pose challenges related to interoperability when different groups from joint forces attempt to coordinate with one another to achieve a mission. Military information systems consist of heterogeneous networks, which are essentially "systems-of-systems."

These systems act across military services, and span political and geographic boundaries. Research is being conducted on formally specifying the necessary interfaces to ensure seamless interoperability among joint networks.

In order to achieve tactical and strategic interoperability, it is desirable to have fully integrated models of the battlefield communication networks which, via simulation,

---

[3]Distributed System Interoperability Perspectives. Position Paper, MITRE Corporation. Bedford, MA, 1996

could be used to identify potential bottlenecks, scenarios for system failure or performance degradation, and also provide an environment for parametric studies for the purpose of optimizing the performance of the networks.

Tactical and strategic interoperability can be facilitated by the use of formal languages, tools, and methodologies. Researchers with the Advanced Telecommunications & Information Distribution Research Program (ATIRP) are designing, prototyping, and evaluating techniques based on standards such as mobile Internet Protocol as well as the next-generation Internet Protocol (Ipv6) to enable interoperability in the US Army's heterogeneous communications network. Emphasis is being placed by the US Army on support for mobility and meeting diverse quality-of-service requirements to enable seamless interoperability of applications across its networks.

Another goal of the US military is to achieve a decisive advantage over its adversaries by moving information reliably to decision makers and weapons operators with security appropriate to its sensitivity. This must be achieved in ways that give the war fighter assurance of the information's authenticity and integrity while weighing potential benefits against potential risks.

The Mobile Ad hoc Networking (MANET) working group and the Internet Engineering Task Force (IETF), among others, investigating mobility and routing protocols, analyze their security strengths and weakness, identify necessary and feasible security extensions for strong authentication, and determine the performance impacts of the identified security extensions. Work is also being conducted on strong authentication mechanisms for use at the link layer on wireless and wired networks.

III. SUMMARY AND COMPARISON OF COMMON ROUTING PROTOCOLS IN WIRELESS NETWORKS

Ad hoc networks, as mentioned before, do not rely on any fixed infrastructure to communicate information across the network. Instead all stations or nodes play the role of router and station at the same time, and thus the importance of routing techniques and protocols used to route data across a set of mobile nodes.

In recent years, a variety of new routing protocols targeted specifically at the ad hoc network environment have been developed. In this section we are going to describe the key features of some of the most important routing protocols such as DSDV, TORA, DSR, AODV, ZRP, and CBRP. These existing routing protocols can be classified as table-driven or on demand routing protocols.

Table driven protocols continuously evaluate the routes within the network, so that when a packet needs to be forwarded, the route is already known and can be immediately used. In contrast, on-demand routing protocols invoke a route-discovery procedure on an on-demand basis. The advantage of the table-driven schemes is that the route information is available when needed, resulting in little delay prior to data transmission at the cost of keeping the routes updated in a highly mobile environment. On the other hand, on demand schemes may produce a significant delay in order to determine a route when route information is needed.

A. DESIRABLE PROPERTIES

In the following paragraph we are going to see some properties that are desirable in ad hoc wireless networks:

a. Distributed Operation

The protocol should of course be distributed. It should not be dependent on a centralized controlling node. This is the case even for stationary networks. The difference is that nodes in an ad hoc network can enter/leave the network very easily and because of mobility the network can be partitioned.

b. Loop Free

It is desirable to get a loop free routing protocol, because it avoids any waste of bandwidth or CPU consumption.

c. Demand Based Operation

To minimize the control overhead in the network and thus not waste network resources more than necessary, the protocol should be reactive. This means that the protocol should only react when needed and that the protocol should not periodically broadcast control information.

d. Power Conservation

The nodes in an ad hoc network can be laptops, and thin clients, such as PDAs that are very limited in battery power and therefore uses some sort of stand-by mode to save power. It is therefore important that the routing protocol supports these sleep-modes.

e. Multiple Routes

To reduce the number of reactions to topological changes and congestion, multiple routes could be used. If one route has become invalid, it is possible that another

stored route could still be valid and thus saving the routing protocol from initiating another route discovery.

f. Quality of Service

Some sort of Quality of Service support is probably necessary to be incorporated into the routing protocol, which has a lot to do with for what these networks will be used.

In the following sections we are going to see the different types of ad hoc wireless routing protocols.

B. DESTINATION-SEQUENCED DISTANCE VECTOR (DSDV)

DSDV is a hop-by-hop distance vector routing protocol requiring each node to periodically broadcast routing updates. The key advantage of DSDV over traditional distance vector protocols is that it guarantees loop-freedom. [Ref. 6]

Basic Mechanisms:

Each DSDV node maintains a routing table listing the "next hop" for each reachable destination, the number of hops to reach the destination and the sequence number assigned by the destination node. DSDV tags each route with a sequence number and considers a route R more favorable than R0 if R has a greater sequence number, or if the two routes have equal sequence numbers but R has a lower metric (i.e., shortest route). Each node in the network advertises a monotonically increasing even sequence number for itself. When a node **B** decides that its route to a destination **D** has become broken, it advertises the route to **D** with an infinite metric and a sequence number one greater than its sequence number for the route that broke (making an odd sequence

number). This causes any node **A** routing packets through **B** to incorporate the infinite-metric route into its routing table until node **A** hears a route to **D** with a higher sequence number. The routing table updates can be sent in one of two ways: full dump or incremental update. A full dump sends the full routing table to the neighbors and could span many packets, whereas in an incremental update only those entries from the routing table that are sent produce a metric change since the last update and must fit in a packet. If there is space in the incremental update packets then those entries may include whose sequence number has changed. When the network is relatively stable, incremental updates are sent to avoid extra traffic and full dumps are relatively infrequent. In a fast-changing network, incremental packets can grow in size so that full dumps will be more frequent.

C. TEMPORARILY ORDERED ROUTING ALGORITHM (TORA)

TORA is a distributed routing protocol based on a "link reversal" algorithm. It is designed to discover routes on demand, provide multiple routes to a destination, establish routes quickly, and minimize communication overhead by localizing algorithmic reaction to topological changes when possible. Route optimality (shortest-path routing) is considered of secondary importance, and longer routes are often used to avoid the overhead of discovering newer routes.

The actions taken by TORA can be described in terms of water flowing downhill towards a destination node through a network of tubes that models the routing state of the real network. The tubes represent links between nodes in the network, the junctions of tubes represent the nodes, and the water in the tubes represents the packets flowing towards the destination. Each node has a height with respect to the destination that is

computed by the routing protocol. If a tube between nodes **A** and **B** becomes blocked such that water can no longer flow through it, the height of **A** is set to a height greater than that of any of its remaining neighbors, such that water will now flow back out of **A** (and towards the other nodes that had been routing packets to the destination via A). [Ref. 10]

Basic Mechanisms:

At each node in the network, a logically separate copy of TORA is run for each destination. When a node needs a route to a particular destination, it broadcasts a QUERY (QRY) packet containing the address of the destination for which it requires a route. This packet propagates through the network until it reaches either the destination, or an intermediate node having a route to the destination. The recipient of the QUERY then broadcasts an UPDATE (UPD) packet listing its height with respect to the destination. As this packet propagates through the network, each node that receives the UPDATE sets its height to a value greater than the height of the neighbor from which the UPDATE was received. This has the effect of creating a series of directed links from the original sender of the QUERY to the node that initially generated the UPDATE.

When a node discovers that a route to a destination is no longer valid, it adjusts its height so that it is a local maximum with respect to its neighbors and transmits an UPDATE packet. If the node has no neighbors of finite height with respect to this destination, then the node instead attempts to discover a new route as described above.

When a node detects a network partition, it generates a CLEAR packet that resets the routing state and removes invalid routes from the network.

Each node "I" contains a "height", which is a quintuple of this form: (T, oid, r, d, I) where:

- T is the logical time of a link failure
- oid is the unique id of the node that defined the reference level
- r is the reflection indicator bit
- d is a propagation ordering parameter
- I is the unique ID of the node

Figure 4 illustrates a route creation process in TORA. As shown in Figure 4a and 4b the QRY packet is created by node C (source) and flooded through the network. An UPD packet propagates back if a route exists to the destination node (Figure 4c, 4d, and 4e). Figure 4e shows that the source node C may have received a UPD each from node A or node G, but since node G gives it lesser height, it retains that height

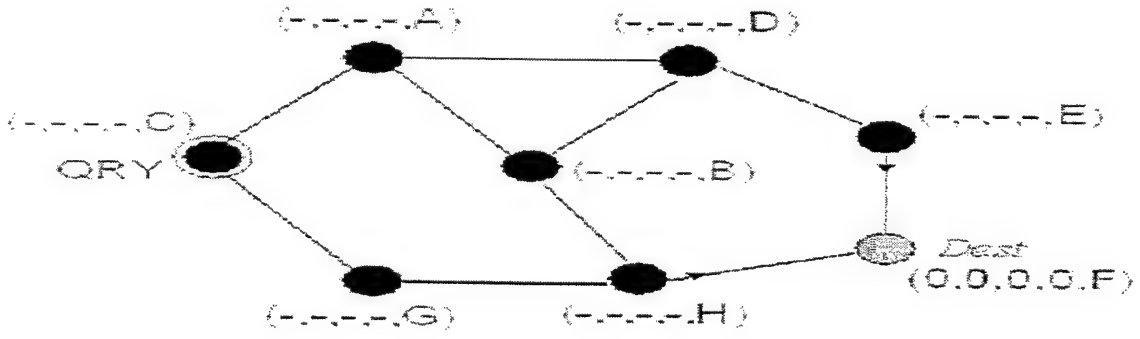

Figure 4.a Route Creation Using QRY Message.

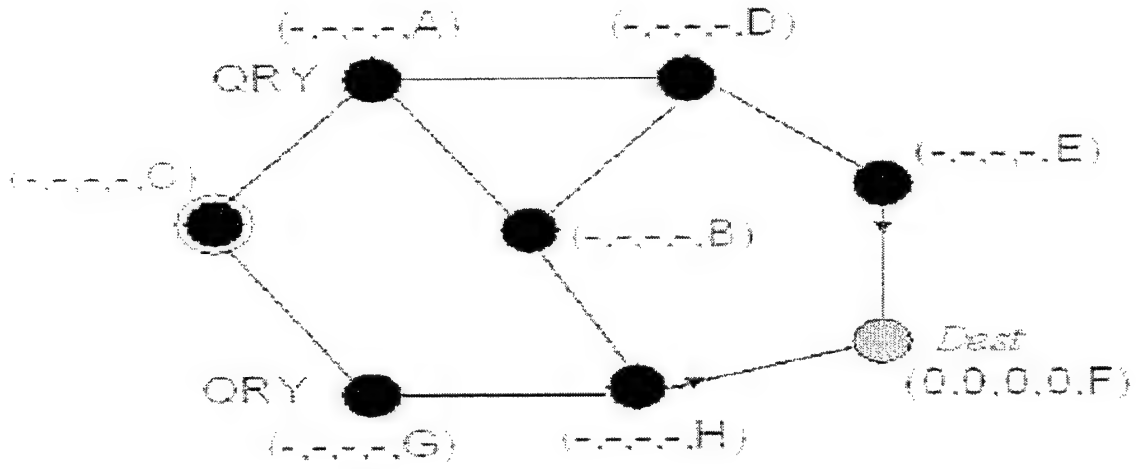

Figure 4.b Propagation of QRY Message Through the Network.

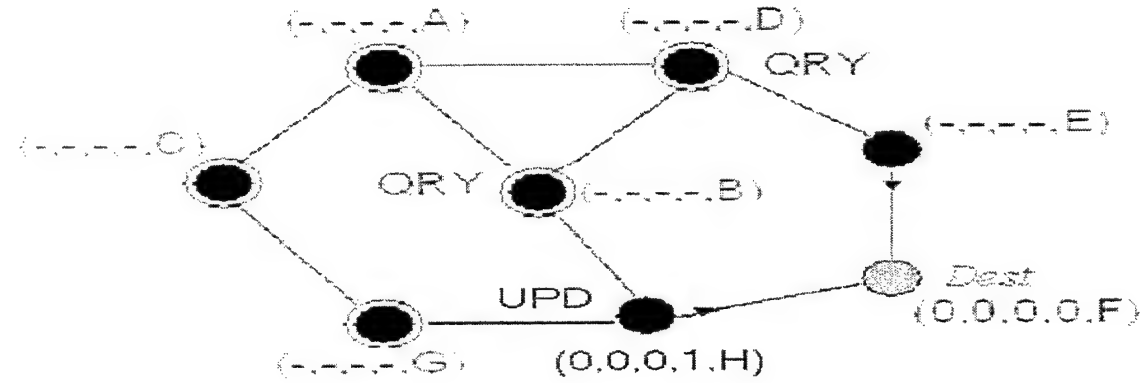

Figure 4.c QRY Propagation and Update of Height of H Node Using UPD Messages.

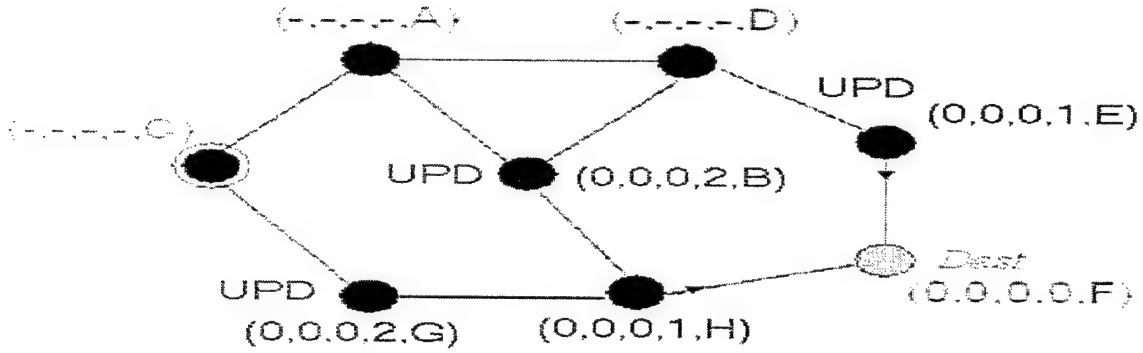

Figure 4.d Update of Height of E, B, and G Nodes Using UPD Messages.

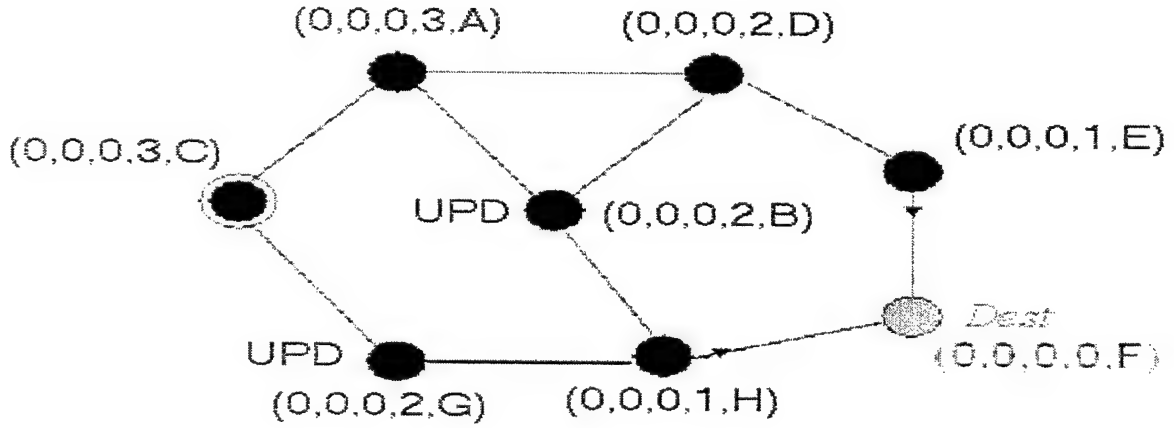

Figure 4.e Update of Height of all Nodes Using UPD Messages.

Figure 4 Routes Discovery in TORA.

D. DYNAMIC SOURCE ROUTING (DSR)

DSR uses source routing rather than hop-by-hop routing, with each packet to be routed carrying in its header the complete, ordered list of nodes through which the packet must pass. The key advantage of source routing is that intermediate nodes do not need to maintain up-to-date routing information in order to route the packets they forward, since the packets themselves already contain all the routing decisions. This fact, coupled with the on-demand nature of the protocol, eliminates the need for the periodic route advertisement and neighbor detection packets present in other protocols. [Ref. 11]

Basic Mechanisms:

The DSR protocol consists of two mechanisms: Route Discovery and Route Maintenance. Route Discovery is the mechanism by which a node **S** wishing to send a packet to a destination **D** obtains a source route to **D**. To perform a Route Discovery, the source node **S** broadcasts a **ROUTE REQUEST** packet that is flooded through the network in a controlled manner and is answered by a **ROUTE REPLY** packet from either the destination node or another node that knows a route to the destination. In order to reduce the cost of Route Discovery, each node maintains a cache of source routes it has learned or overheard, which it aggressively uses to limit the frequency and propagation of ROUTE REQUESTs.

Route Maintenance is the mechanism by which a packet's sender **S** detects if whether the network topology has changed such that it can no longer use its route to the destination **D** because two nodes listed in the route have moved out of range of each other. When Route Maintenance indicates a source route is broken, **S** is notified with a

ROUTE ERROR packet. The sender **S** can then attempt to use any other route to **D** already in its cache or can invoke Route Discovery again to find a new route.

E. AD HOC ON-DEMAND DISTANCE VECTOR (AODV)

AODV is essentially a combination of both DSR and DSDV. It borrows the basic on-demand mechanism of Route Discovery and Route Maintenance from DSR, plus the use of hop-by-hop routing, sequence numbers, and periodic beacons from DSDV. [Ref. 10]

Basic Mechanisms:

When a node **S** needs a route to some destination **D**, it broadcasts a ROUTE REQUEST message to its neighbors, including the last known sequence number for that destination. The ROUTE REQUEST is flooded in a controlled manner through the network until it reaches a node that has a route to the destination. Each node that forwards the ROUTE REQUEST creates a *reverse route* for itself back to node **S**.

When the ROUTE REQUEST reaches a node with a route to **D**, that node generates a ROUTE REPLY that contains the number of hops necessary to reach **D** and the sequence number for **D** most recently seen by the node generating the REPLY. Each node that participates in forwarding this REPLY back toward the originator of the ROUTE REQUEST (node S) creates a *forward route* to **D**. The state created in each node along the path from **S** to **D** is the hop-by-hop state. Each node remembers only the next hop and not the entire route, as would be done in source routing.

In order to maintain routes, AODV normally requires that each node periodically transmit a HELLO message, with a default rate of once per second. Failure to receive three consecutive HELLO messages from a neighbor is taken as an indication that the

link to the neighbor in question is down. Alternatively, the AODV specification briefly suggests that a node may use the physical layer or link layer methods to detect link breakages to nodes that it considers to be neighbors.

When a link goes down, any upstream node that has recently forwarded packets to a destination using that link is notified via an UNSOLICITED ROUTE REPLY containing an infinite metric for that destination. Upon receipt of such a ROUTE REPLY, a node must acquire a new route to the destination using Route Discovery as described above.

F. ZONE ROUTING PROTOCOL (ZRP)

In the Zone routing protocol, each node has its own "routing zone" which includes the nodes whose distance (hops) is at most some predefined number. Each node is required to know the topology of the network within its routing zone only, and route updates are propagated only within the routing zone. A proactive protocol such as DSDV is used within the routing zone to learn about its topology. [Ref. 10]

In order to discover a route to an out-of-zone node, a reactive protocol such as DSR is used. Note that ZRP exhibits hybrid behavior of proactive and reactive protocols through the use of the zone radius. For a large zone radius, ZRP is more proactive, and for a small zone radius, ZRP is more reactive.

The routes discover protocol used in ZRP is illustrated in Figure 5. Let us assume that the source **A** wants to find out the route to destination **E**. A first verifies that E is out of its zone. It then sends a query packet to all the nodes on the periphery of its zone, that is, **B** and **C**. Upon receiving a query packet, each of these nodes appends its address to the query packet and forwards it to its peripheral nodes since **E** is not in its routing zone.

In particular, **B** forwards the query packet to **D**, which recognizes **E** as its zone member. **D** then sends a reply packet, which includes **A-B-D-E** route.

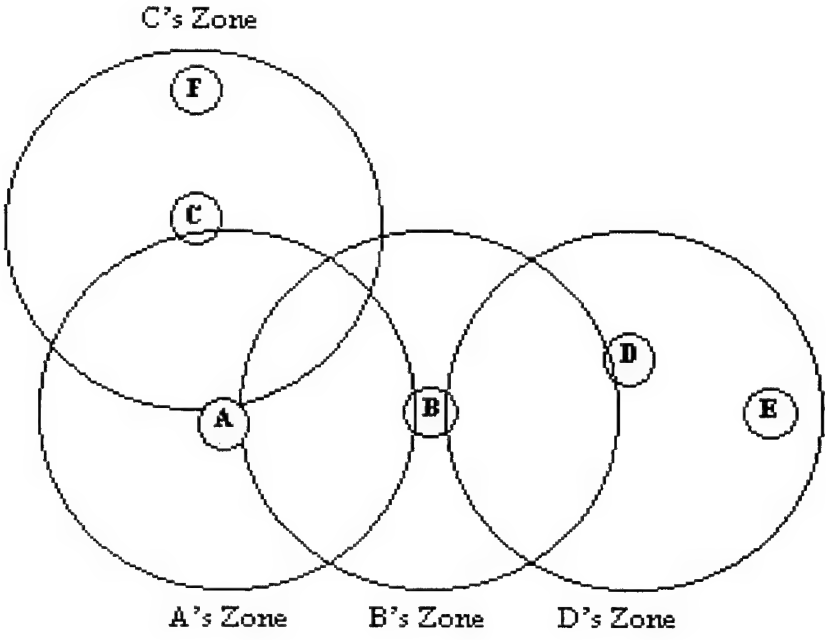

Figure 5. Example of Zone Routing Protocol.

The advantage of ZRP is that it significantly reduces the communication overhead as compared to the pure proactive protocols since in ZRP each node needs to know the topology of its zone only. In addition, ZRP discovers routes faster than the pure reactive protocols since in ZRP only the peripheral nodes are queried in the route discovery process. It is also noted that the ZRP path, which consists of nodes spaced approximately by distance of zone radius, is more stable than the full path, which consists of all the nodes between the source and the destination. This is because there are some topological changes, which affect the full path, but not the ZRP path.

G. CLUSTER BASED ROUTING PROTOCOL (CBRP)

Cluster Based Routing Protocol (CBRP) is a routing protocol designed for use in mobile ad hoc networks. The protocol divides the nodes of the ad hoc network into a number of overlapping or disjoint clusters in a distributed manner. A cluster head is elected for each cluster to maintain cluster membership information. Inter-cluster routes are discovered dynamically using the cluster membership information kept at each cluster head. By clustering nodes into groups, the protocol efficiently minimizes the flooding traffic during route discovery and speeds up this process as well. Furthermore, the protocol takes into consideration the existence of unidirectional links and uses these links for both intra and inter-cluster routing.

Basic Mechanisms:

Route Discovery is the mechanism whereby a node S, wishing to send a packet to a destination D, obtains a source route to D. The way S finds a route (or multiple routes) to D is also done by flooding. However, because of the clustering approach, the number of nodes that are used are much less in general.

In Route Discovery, cluster heads are flooded in search of a source route. To perform Route Discovery, the source node S sends out a Route Request Packet (RREQ), with a recorded source route listing only itself initially. Any node that forwards this packet will append its own ID in this RREQ. Each node forwards a RREQ packet only once and it never forwards it to a node that has already appeared in the recorded route. In CBRP, the RREQ will always follow a route with the following pattern to reach destination D: S,CH1,G1,CH2,G2,G3,CH3 ..... D

A detailed description of how this is achieved is presented below.

Source always unicasts RREQ to its cluster head, say CH1. Each cluster head will unicast RREQ to each of its bi-directionally linked neighbors, which have not yet appeared in the recorded route through the corresponding gateway. This process continues until the target is found or until another node that can supply a route to the target is found.

When the target of the Request, node **D**, receives the RREQ, **D** may choose to memorize the reversed source route to **S**. Node **D** then copies the recorded source route into a Route Reply packet (RREP), which it then sends back to the initiator of the Route Request (e.g., node **S**) by reversing the recorded route and putting it in the IP header of the Route Reply packet. The recorded route gives the complete information about the SEQUENCE OF CLUSTERS source should traverse in order to reach destination **D**. While forwarding the Route Reply, intermediate cluster heads modify the IP header of the packets, and substitute the inter-cluster incoming links to inter-cluster outgoing links. Each intermediate cluster head also modifies the recorded route in the Route Reply packet to optimize the recorded route as much as possible using its knowledge of the cluster topology and inter-cluster gateway information.

An example of such optimization is to connect two gateway nodes by an intra-cluster link that does not go through the cluster head. All source routes learned by a node are kept in a Route Cache, which is used to further reduce the cost of Route Discovery. When a node wishes to send a packet, it examines its own Route Cache and performs Route Discovery only if no suitable source route is found in its cache.

H. AD HOC ROUTING PROTOCOLS COMPARISON

Each of the protocols has its pros and cons. Table 1 presents a comparison of these different routing protocols.

	DSDV	TORA	DSR	AODV	ZRP	CBRP
Loop free	Yes	No, short lined loops	Yes	Yes	Yes	Yes
Multiple routes	No	Yes	Yes	No	No	Yes
Distributed	Yes	Yes	Yes	Yes	Yes	Yes
Reactive	No	Yes	Yes	Yes	Partially	Partially
Unidirectional link support	No	No	Yes	No	No	Yes
Periodic broadcasts	Yes	Yes	No	Yes	Yes	Yes
QoS support	No	No	No	No	No	No
Power conservation	No	No	No	No	No	No

Table 1. Comparison of Routing Protocols.

As it can be seen from Table 1, none of the protocols support power conservation or Quality of Service.

DSDV is the only proactive protocol in this comparison. It is also the protocol that has most in common which traditional routing protocols in wired networks. DSDV will probably be good enough in networks, which allows the protocol to converge in reasonable time. This, however, means that the mobility cannot be too high, which is why the authors of DSDV designed the AODV, which is a reactive version of DSDV. The reactive approach in AODV has many similarities with the reactive approach of DSR.

They both have a route discovery mode that uses request messages to find new routes. The difference is that DSR is based on source routing and will learn more routes than AODV. DSR also has the advantage that it supports unidirectional links. DSR has, however, one major drawback and it is the source route that must be carried in each packet. This can be quite costly, especially when QoS is going to be used.

ZRP and CBRP are two very interesting proposals that divide the network into several zones/clusters. This approach is probably a very good solution for large networks. Within the zones/clusters they have a more proactive scheme and between the zones/clusters they have a reactive scheme that has many similarities with the operation of AODV and DSR. They have, for instance, a route discovery phase that sends request through the network. The difference between ZRP and CBRP is how the network is divided. In ZRP all zones are overlapping and in CBRP clusters can be both overlapping and disjoint.

None of the presented protocols are adaptive. This means that the protocols do not take any smart routing decisions when the traffic load in the network is taken into consideration. As a route selection criteria the proposed protocols use metrics such as the shortest number of hops and quickest response time to a request. This can lead to the situation where all packets are routed through the same node even if there exist better routes where the traffic load is not as large.

I. TCP AND WIRELESS ROUTING PROTOCOLS

TCP/IP is the standard networking protocol on the Internet and the most widely used transport protocol for data services like file transfer, email, and other types of applications. TCP is an end-to-end protocol designed for the wireline networks

characterized by negligible random packet losses. Use of TCP, as the transport protocol over the wireless links, is not an efficient solution due to the different characteristics of the wireline and the wireless links. This is because any packet loss over the wireline links is mainly on account of congestion, unlike wireless links where packet losses can result both due to congestion and random losses. Since TCP does not distinguish between congestion losses and losses that are due to route failure, the throughput of a TCP connection over a wireless link suffers, and degrades significantly when nodes move. In spite of this problem, the TCP protocol is still used to transfer data over the wireless link. However, research is underway to come up with an efficient transport protocol over wireless communication that can replace TCP. [Ref. 12]

J. PROBLEMS ASSOCIATED WITH ROUTING PROTOCOLS IN AD HOC WIRELESS NETWORKS

Although, several routing schemes have been proposed in ad hoc wireless networks, most of them are modified extensions of existing link state or distance vector based routing protocols.

In ad hoc mobile network where hosts are acting as routers and have power and bandwidth constraints, conventional routing protocols, which employ periodic broadcast, are unlikely to be suitable. Consequently, there is a need for simple, bandwidth efficient and robust routing protocol for ad hoc mobile networks that can assure good quality of service (QoS).

As stated before, in an ad hoc wireless network, all communication is done over wireless media, typically by radio through the air, without the help of wired base stations.

Since direct communication is allowed only between adjacent nodes, distant nodes communicate over multiple hops. The quality-of-service (QoS) routing in an ad hoc network is difficult because the network topology may change constantly, and the available state information for routing is inherently imprecise. Therefore, there is a need to routing algorithms with a QoS routing scheme that selects a network path with sufficient resources to satisfy a certain delay (or bandwidth) requirement in a dynamic multihop mobile environment

In fact, QoS is a very important issue in ad hoc wireless networks. It refers to traffic-dependant performance metrics, such as bandwidth, end-to-end latency, or the likelihood of message loss that a connection must have to tolerate the type of data transmitted. A network's admission-control mechanisms must be present to be invoked whenever a new connection is initiated. This mechanism plays a very important role in assuring that QoS requirements will be met, and in aborting any connection otherwise.

Regarding the routing protocols scheme used in ad hoc wireless networks, it is obvious that it is difficult to provide QoS in such an environment. The overhead of QoS routing in an ad hoc network is likely to be higher than that in a wireline network because the available state information is less precise and the topology changes in an unpredicted way.

The provision of QoS relies on resource reservation. Data packets of QoS connection are likely to flow along the same network path on which the required resources are reserved.

Routing is the first step in resource reservation and consists of:

- Selecting network paths that have sufficient resources to meet the QoS requirements to any admitted connection
- Achieving global efficiency in resource utilization

Several routing algorithms have been developed to meet the first bullet. Lin and Gerla[4] proposed an algorithm that introduces the bandwidth constraints to traditional routing protocols. This routing algorithm keeps track of the shortest path for all bandwidth values. To find out about the paths that are needed to meet the QoS, each node periodically broadcasts to its neighbors the {bandwidth, hop distance} pairs for the preferred paths to each destination. If a node receives a packet with a bandwidth request, which cannot be satisfied by the currently available paths to the intended destination, it drops the packet without acknowledgment (ACK). Eventually, the sender will reroute the call on other path.

Chen and Nahrstedt.[5] proposed a better algorithm for satisfying QoS requirements in ad hoc wireless networks. They proposed a ticked-based distributed QoS routing scheme for ad hoc networks. In fact, the existing single path routing algorithms have low overhead but do not have the flexibility of dealing with imprecise state information. On the other hand, the flooding algorithms can handle information imprecision but have high overhead.

---

[4] Lin, C. R.., "Real-Time support in Multihop wireless Networks," Wireless Networks 5, 1999, 125-135.

[5] Chen S., Nahrstedt K., "Distributed Quality of Service Routing in Ad Hoc Networks," IEEE Journal.

The proposed ticket-based probing scheme achieves a balance between single-path routing algorithms and the flooding algorithms. It does multihop routing without flooding. The basic idea is to achieve an optimal performance with low overhead by using a limited number of tickets and making intelligent hop-by-hop path selection..

The ticket-based probing algorithm consists of the following: a ticket is the permission to search one path. The source node issues a number of tickets based on the available state information. There are more tickets issued for connections with tighter requirements. Whenever a node wants to communicate with another node, a set of routing messages are sent from the source toward the destination to search for a low-cost path that satisfies the QoS requirements. Each probe (routing messages) is required to carry at least one ticket. At an intermediate node, a probe with more than one ticket is allowed to be split into multiple ones; each is searching a downstream sub path. The maximum number of probes at any time is bound by the total number of tickets. Since each probe searches a path, the maximum number of paths searched is also bound by the number of tickets. See Figure 6 for an example. Upon receipt of a probe, an intermediate node decides, based on its state, whether the received probe should be split, and to which neighbor nodes the probe(s) should be forwarded. The goal is to collectively utilize the state information at the intermediate nodes to guide the limited tickets along the best paths to the destination, so that the probability of finding a low-cost feasible path is maximized.

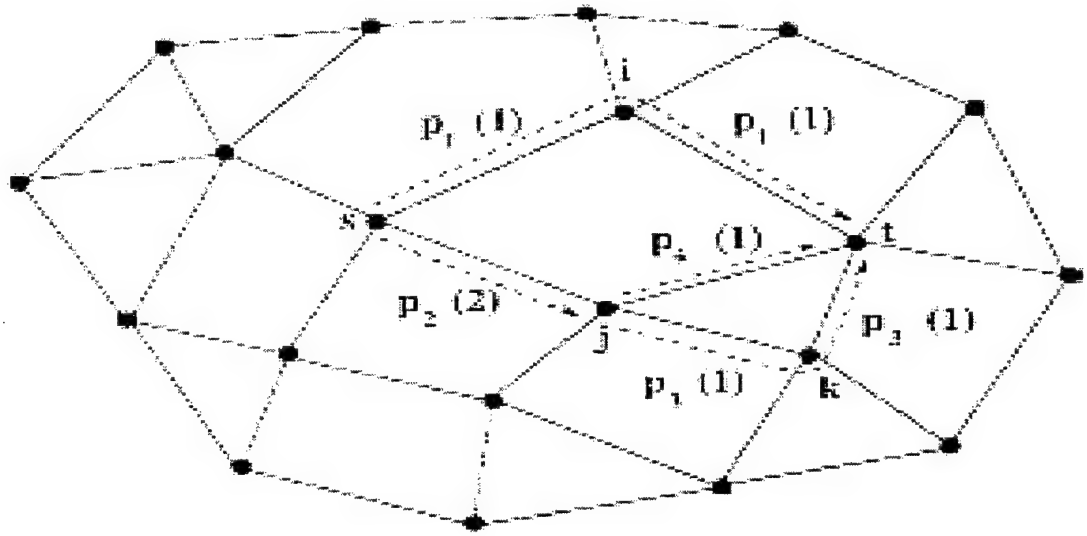

Figure 6. Ticket-Based Probing Algorithm: Two probes, p1 and p2, are sent from s (source node). The number in the parentheses following a probe is the number of tickets carried in the probe. At node j, p2 is split into p3 and p4, each of which has one ticket. There are at most three probes at any time. Three paths are searched, and they are:

s → i → t, s → j → t, and s → j → k → t

In the same token, when a connection request arrives at the source node, a certain number N of tickets are generated, and probes are sent toward the destination t. Each probe carries one or more tickets. Since no new tickets are allowed to be created by the intermediate nodes, the total number of tickets is always N, and the number of probes is at most N at any time. When a node receives a probe p with N (p) tickets, it makes at most N (p) copies of p, distributes the received tickets among the new probes, and then forwards them along to selected outgoing links toward t. Each probe accumulates the delay of the path it has traversed so far. A probe can proceed only when the accumulated delay does not violate the delay requirement. Hence, any probe arriving at the destination detects a feasible path, which is the one it has traversed.

K. REROUTING IN AD HOC WIRELESS NETWORKS

Due to the mobility of nodes, paths are subject to being broken frequently. Rerouting is commonly used to deal with this kind of problem.

In an ad hoc network, there are a number of situations where rerouting is desired. First, the network topology may change as new nodes join in the network and existing nodes move or leave the network. Rerouting helps to tolerate the network dynamics by adapting the routing paths periodically according to the changing topology. More importantly, when a routing path is broken, rerouting can be used to reestablish the connection along a new path. Second, the routes of the connections are typically selected based on the network resource availability at the times when the requests arrive. Long paths are often assigned when resource contention occurs. However, as the network topology changes and connections are established or torn down upon completion, the network state changes locally and globally. Routes with light (heavy) traffic at the beginning may become congested (lightly loaded) later. Shorter paths for some connections may become available. Rerouting helps to balance the network traffic on the fly and improves the resource efficiency, which is especially important in an ad hoc network where resources are scarce. Rerouting can be done periodically and/or upon triggering when a broken path is detected. It should not be done too frequently in order to avoid excessive overhead and the oscillation of shifting the traffic from one part of the network to another. However, it should also be noted that, compared to the contiguous traffic of a typical voice connection, the rerouting overhead is relatively small as long as it is not done too frequently.

IV. REQUIREMENTS FOR SEMANTIC INTEROPERABILITY AMONG WIRELESS AD HOC NETWORKS

A. OVERVIEW

Realizing interoperability is a challenge, especially across large-scale, distributed systems. However, there are best practices used to enable a higher degree of interoperability, such as adapting:

- Common operating environments (e.g., the US DoD COE)
- Standards on the interfaces (e.g., use of CORBA or DCE)
- Common interfaces
- Common architecture

These best practices cannot be always applied to all the possible situations. In the case of joint forces using various types of network architectures and functionalities, the interoperability is an issue. Thus, in this case, we must address interoperability from the perspective of different (or heterogeneous) systems that wish to operate and communicate in a transparent environment.

> A mission application that needs to interoperate with another mission application (from another system environment) also needs to interoperate with its own distributed system components. Those distributed system components, which formulate the operating environment (OE) for the mission application, need to interoperate with their counterparts' operating environment that supports the remote mission application operation environment[6].

---

[6]Distributed System Interoperability Perspectives. Janis R. Putman. MITRE Corporation Bedford, MA, 1996

A semantic interoperability relationship needs to be established between the two OEs in order for both applications to communicate in a seamless way.

In this chapter, we define the components of semantic interoperability services and the associated requirements, in addition to the role of middleware and metadata in providing for semantic interoperability within an ad hoc wireless network.

B. COPONENTS OF SEMANTIC INTEROPERABILITY IN AD HOC WIRELESS NETWORKS

Military wireless ad hoc networks, in the case of joint forces using various types of network architectures or in the case of the same forces using different operating environments, need to interoperate in order to accomplish their missions. Flexibility in interconnecting distributed computers from different operating environment is not easily achieved because there are challenges to be addressed in

- Finding and invoking services
- Passing arguments
- Interpreting the received arguments

Distributed Object Managers (**DOMs**) have been proposed as a way to resolve the first two difficulties. DOMs allow a client application to access computing resources independently of their location. The client does not need to know the location of the server. An example of DOMs is the Common Object Request Broker Architecture (**CORBA**). CORBA provides a standard interface called Object Request Broker (**ORB**), through which clients can request that operations on objects be performed.

In order to address the challenges posed for achieving semantic interoperability, a set of Semantic Interoperability Services (**SIS**) is needed to be put in place. Semantic

interoperability is needed when the sending application from the same network or another network has different expectations than the receiving application. In this case, semantic interoperability service performed between the sending of an argument value **v** and the receiving of the converted value **v'** is called *mediation*. *Mediation* involves the following operations:

- Determining the expectations that the two applications have for value **v**
- Choosing how to convert value **v** to **v'** (or raising an expectation if conversion is not possible)
- Choosing where and when the conversion shall occur, and
- Arranging for the execution of both the conversions and the user-involved function

These mediation activities consist of the following functional areas:

- A collection of argument describers (one for each server function or client request)
- A library of conversion functions
- A planner that produces a conversion strategy
- A request broker

1. Argument Describers and Descriptors

One can express an application's assumptions about an argument's meaning. The role of the argument describer is to determine the assumptions. The representation used to convey these assumptions is called an argument descriptor. The knowledge required by an argument describer must be supplied either by the application developer or by the

semantic interoperability services. In order to specify a semantic interoperability service one must determine the structure of the argument descriptor, how an argument describer determines the descriptor for a given argument value, and whether all arguments must have descriptors.

a. Structure of an Argument Descriptor

> A semantic interoperability service needs a descriptor format that is widely adopted; it should be easy to understand, extend, and process. Property value lists can be used to specify and define the property names and their values for a given application[7].

For example, consider two applications that are communicating over the network, exchanging documents via a request argument. Assume that they agree on the basic meaning of the argument (e.g., that the indicated file is to be edited and displayed). The partners (sender and receiver) might need to mediate detailed information about the document to be exchanged. This information can contain the vendor format (e.g., WordPerfect, PostScript, MacWrite), and whether the document is uncompressed or compressed.

Each partner must provide routines that map all necessary format information to and from the descriptor, and all partners must use compatible encoding schemes. In the document example, if there are only the three types of documents that are supported, then the applications could use six string values to encode the possible formats, say "WP/U", "WP/C", "MW/U", "MW/U", "PS/U", and "PS/C". Unfortunately,

---

[7]Description, Conversion, and Planning for Semantic Interoperability. A. Rosenthal and E. Sciore. MITRE Corporation Bedford, MA, 1995

such encoding hides information that an SIS could exploit.

This situation can be modeled more clearly by structuring the argument descriptor as properties. A property is a category of information that describes an aspect of the argument's semantics. This aspect may be essential in deciding how to use the data, or may just describe representational details (e.g., units, data types) that conversion functions can hide. An argument descriptor is a set of property- property value pairs.

In the above example, a compressed WordPerfect document could have the one property argument descriptor {(docFmt, "WP/C")} or the two-property argument descriptor {(product, "WP"),(compressionStatus, "Yes")}. The latter descriptor is more expressive. Description and conversion tasks are now decomposable, which simplifies their administration and use. With the argument descriptors model, conversion functions can be written to handle one property at a time, for example, (units, miles), (units, km), (datatype, string),(datatype, float), (compressionStatus, "Yes"), (compressionStatus, "No"). Argument descriptors can have a different level of complexity depending on the type of the property list that might be fixed or extensible.

When new applications are made accessible through a distributed system, these applications may make distinctions that were not anticipated when the encoding and property names were defined. If the list of property-names is fixed, then it can be difficult to accommodate such additions. An extensible list of property names can make it easier for applications from different networks to adjust their argument descriptor. For example, a client-server order entry system in the U.S might implicitly understand all currency values to be in US dollars. When a new user in another country gets added to the system,

all relevant argument describers need to be identified and then modified to support the property currency. This is much simpler when the property lists are extensible.

b. When is Each Argument Descriptor Determined?

In each network communication, such as a client request and a server reception of that request, it is necessary for each partner to understand the assumptions being used by the other partner. An argument describer is a kind of function that takes two inputs, a request and an argument of the request, and returns the descriptor for that argument. The describer must determine what properties are in the descriptor, and what values the properties have. It is not always easy to get those properties out of some applications, because either robustness of the application is not a concern, or the language provides no means for describing the assumptions. The issues of extracting explicit information are discussed later. These difficulties are serious because they present several problems in determining assumptions underlying a request. However, the client applications, in general now, are aware of their assumption when interacting with a server (from the same or different operating environment).

We now discuss options for where and when each property name or value may be determined.

- Property information may be determined by rules stored in a knowledge base, for example a system reporting traffic violation in the US reports all speed properties with property value "miles per hour," whereas a system from another country might supply the same property value but in "kilometers per hour"

- Property information may be stored with the value (e.g., the suffix of the file name)
- Property information may be negotiated when the applications first connect. For example, the hypertext transport protocol (HTTP) specifies that one or both participants provide a list of the formats they support.
- The property value may be generated at request time

Moreover, it is necessary to establish how partners (sender and receiver) agree on the meaning of property names and values. The meaning of an argument descriptor needs to be understandable to all of the participants. Thus, the property names and values for each application must be understood by both applications' argument describers. It is necessary to avoid homonyms and to minimize and exploit synonyms.

- Homonyms[8] present problems when two applications use the same property name or property value to denote different things. For example, describers from a land application and a naval application might both use mile, but would mean respectively conventional and nautical miles (roughly, 5300 versus 6100 feet). Such homonyms must be prevented because they lead to wrong results.

- Synonyms are different names for the same thing, meaning that each application describer is using a different property name for the same argument. An approach avoiding the use of synonyms is to restrict the argument describers to using a controlled vocabulary, or a set of concepts

[8]This is also known as semantic overloading of terms

that are assumed, and known to all participants and provide a point of reference for all descriptions.

2. Conversion Functions

In order to solve many of the conversion problems that might arise between two systems that need to interoperate, it is necessary to have different kinds of conversion functions that are capable of performing particular conversion operations. These functions are stored in a conversion library.

A conversion function is a routine that is capable of converting a sender's value **v** into a new value **v'** that is semantically consistent with "v" but whose descriptor has different property values. These functions must guarantee that the result is consistent with the input, especially for complex types, because the conversions may be lossy (e.g., omitting information that cannot be represented in the result format).

A conversion function can be defined as a map between one argument descriptor and another. A property conversion function takes as input an argument value **v**, an argument descriptor for the sender, and the desired property value for the receiver. Its output is the desired value **v'**. For example, an application that takes GIF files as input and produces JPEG files, as output is a FormatConvert conversion function. Another example would be a function that converts miles to km. The property values described in the argument descriptor for a given application that needs to request another application in a different operating environment may be computed by the conversion function, rather than being specified in the call.

It is necessary to determine before runtime whether the conversion plans will execute correctly: correct behavior is predicated on correctly specifying the preconditions and post-conditions on the conversions.

a. Specification of Preconditions and Post-Conditions

The semantic interoperability service needs detailed information about the argument types and a set of both preconditions and post-conditions for each conversion function. The planner needs to understand the conversion's preconditions on the value **v** and its property values. Post-conditions are also needed, which are the specification of the descriptor for the value after the conversion process is complete, as either a constant or a function of the incoming descriptor. For example, a precondition can specify that the convertibility of one property depends on the other properties, such as if the sender's value has the format {(encryption, RSA), (datatype, integer), (units, meters)}. It is meaningless to apply a units conversion before the value is decrypted. In this kind of conversion, one can use preconditions that reference any argument or descriptor in the request and can specify that there are property values (e.g., compression, encryption) that need to be converted prior to any other conversion.

b. Minimization of Conversion Loss

Conversion functions should minimize the degree of information loss. Information loss may include loss of precision (e.g., truncation trailing digits), granularity, certain kinds of information (e.g., labeling of keywords in a document), and so on. If the level of information loss exceeds a certain threshold value, then the application that depends on the conversion might behave in an unintended manner.

3. The Planner

The planner is the component that compares the client and server argument descriptors for each argument in a request and produces a conversion strategy, which is a sequence of calls to conversion functions. The planner's power and design are dependent on the expressiveness of argument descriptors and the contents of the conversion library. The planner determines the appropriate way to convert from one property value to another and it creates a conversion plan in the case of the existence of different properties.

a. Planning to Convert From One Property Value to Another

In this simplest case, the planner checks whether there exists a direct conversion. If it fails to find one, the conversion fails. An alternative strategy is the one that converts to and from a common interchange format. A standard interchange format operates by selecting a value **v** as the standard through which the communication should pass.

A common interchange format depends on the richness of the conversion function library. We can have the following possibilities:

- Full convertibility: in this case we have a standard representing the property P that is called the standard interchange value **v**. Every other format is convertible to and from this standard.
- Limited convertibility: in the previous case (i.e., full convertibility) we still use a standard format, but the conversion library is incomplete because there may be insufficient resources available to write all of the

desired conversions. For example, it may be that someone imports documents (e.g., for display) but never exports them.

- Conversion between nonstandard formats: in this case there is no use of a standard interchange format. Instead, specific conversion functions are used to convert from one property value to another.

A planner that combines the use of the simplest planner strategy and the use of common interchange format offers much more power to meet users' needs than otherwise would be the case.

b. Planning for Multi-Property Conversion

Next we consider the role of the planner in converting between multi-property argument descriptors. There exists a requirement for the planner that all properties be independent of each other in the sense that converting one property does not affect the convertibility of any other properties. Furthermore, the planner must be able to handle interference between properties. Successfully meeting the second requirement is dependent on the planner's ability and effectiveness at identifying desirable behaviors for conversion functions. For example, property names might be orderable by resistance to interference so that later properties (e.g., units or datatype) do not affect the conversion of earlier properties (e.g., encryption). Another approach would be to specify a base value for each non-independent property. Such a value would not interfere with any other conversions (e.g., once a document is decrypted, all other conversions can apply). Therefore, if these resistance-ordering and base-value assumptions are known, the planner can create a conversion plan to convert the most interference-resistant property to

its base value, perform the other conversions recursively, and then convert the property to its target value.

For example, suppose that the sender format is {(units, inches), (encryption, RSA)} and the receiver wants {(units, feet), (encryption, RSA)}. The planner at this level would be expected to determine that when the property 'encryption' has the base value "none" then the planner should respond by generating a plan that first decrypts the value, then performs the units conversion, and finally re-encrypts the value.

An application itself may provide some degree of planning and conversion. For instance, Microsoft Word 5.0 can write documents (with some information loss) in roughly a dozen formats. Upon reading, it checks file descriptors or headers and can then read a similar number of formats. However, the level of mediation functionality supported by legacy applications is typically low. For example, the descriptors produced by applications often consist of just one property, and conversions must be taken directly from a library rather than be dynamically composed.

Figure 7 summarizes the set of operations needed for two applications from different operating environments to exchange information in a seamless way.

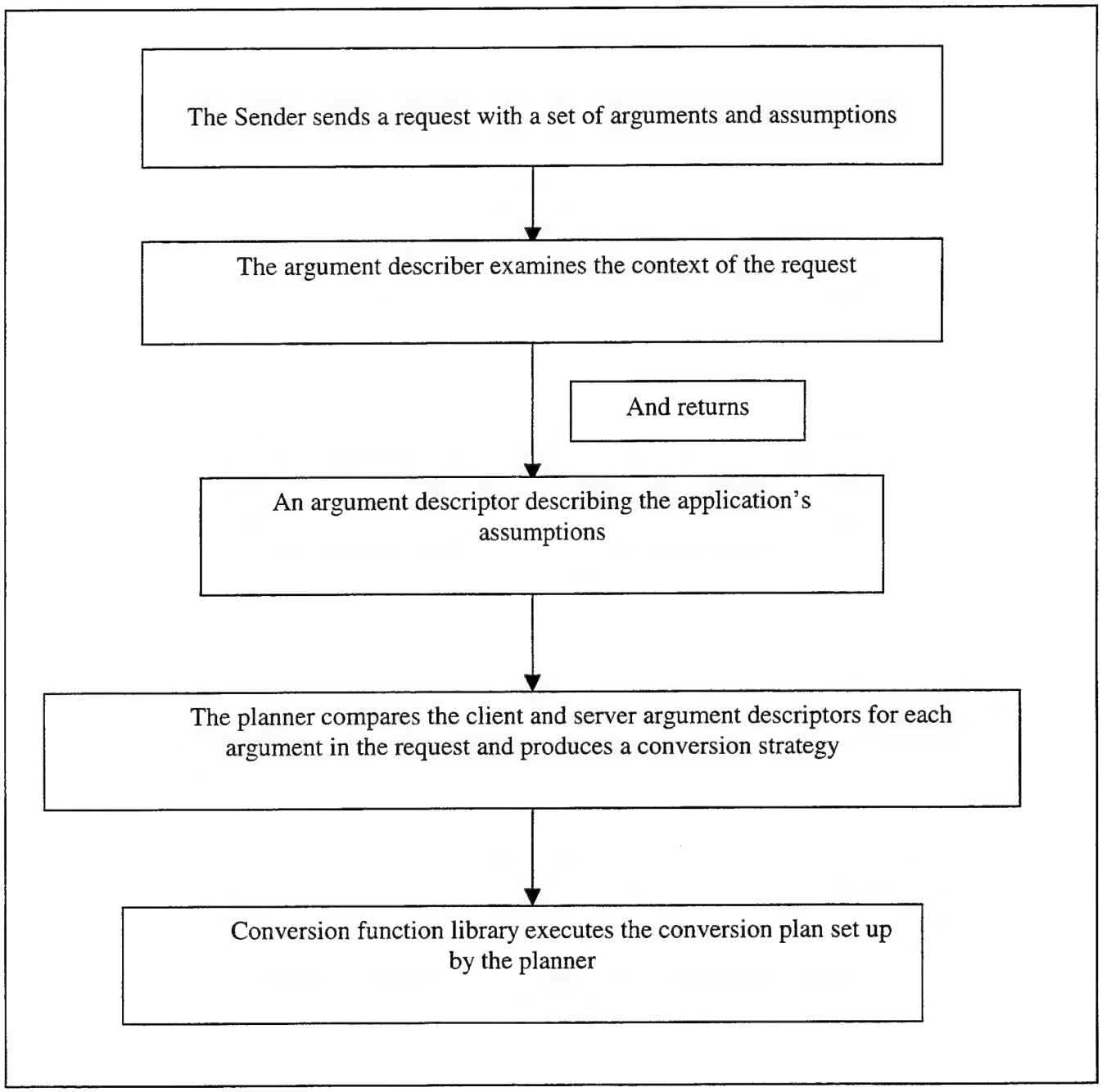

Figure 7. Steps in the Conversion Planning Process and the Execution of the Plan.

4. The Object Request Broker (ORB)

The role of the Object Request Broker (ORB)[9] is to assist a client in invoking a method of a remote object. The Common Request Broker Architecture (CORBA) specification defines how ORBs from different vendors can communicate using a common protocol. ORBs promote interoperability of distributed object systems because they enable users to build systems by piecing together objects from different vendors that communicate with each other via the ORB. The implementation details of the ORB are generally not important to developers building distributed systems. The developers need only be concerned with the details of the object's interface.

ORB technology promotes the goal of object communication across machine, software, and vendor boundaries. The relevant functions of an ORB technology are

- Interface definition
- Location and possible activation of remote objects
- Communication between clients and object's such as shown in Figure 8

[9] The object management group (OMG) was formed in April 1989. In 1991, the OMG announced its adoption of the CORBA specification.

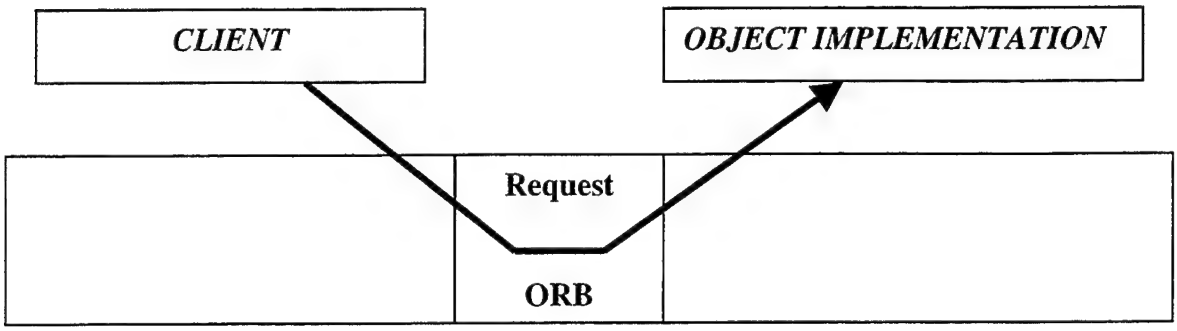

Figure 8. ORB Request.

An object request broker acts as a kind of telephone exchange. It provides a directory of services and helps establish connections between clients and these services, as shown in Figure 9.

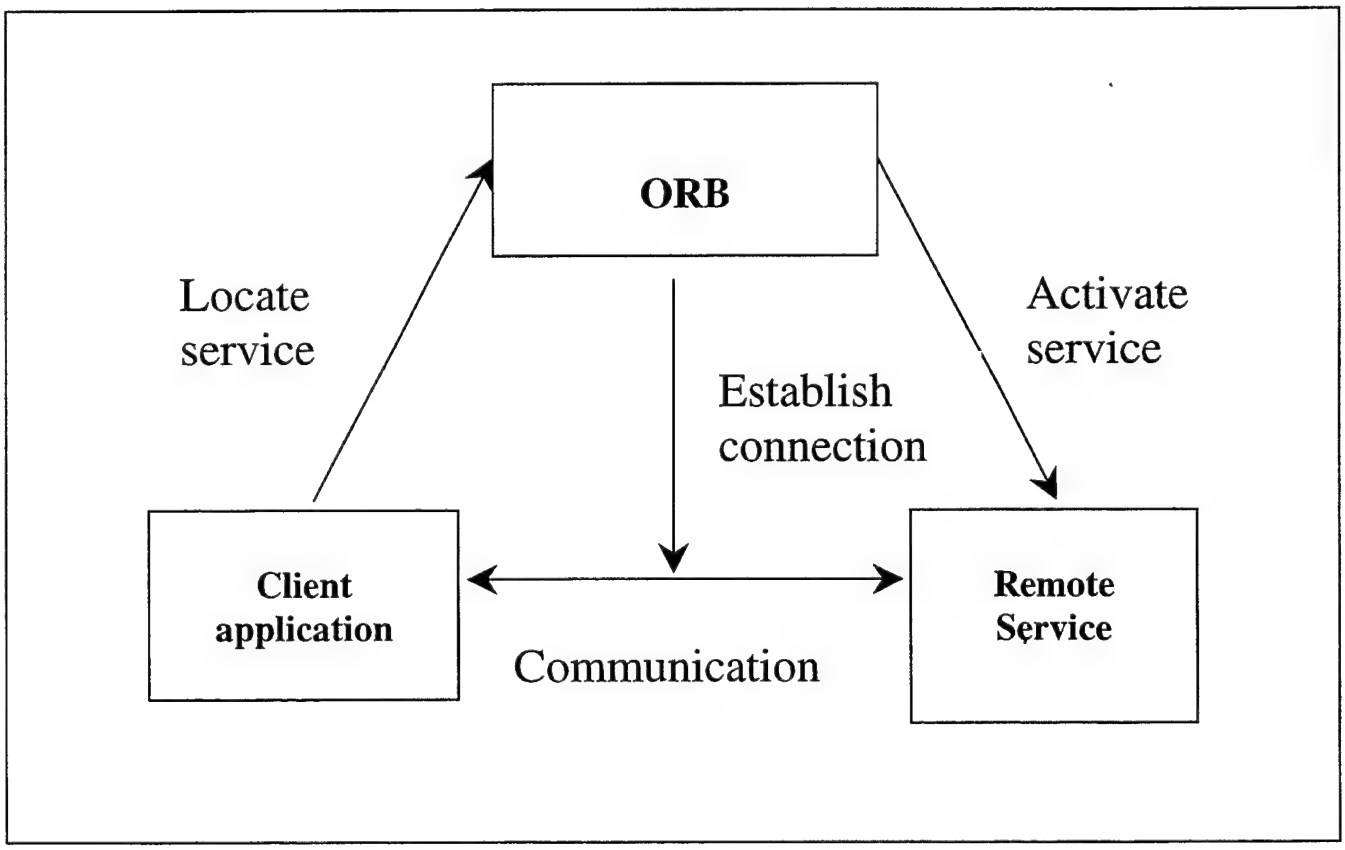

Figure 9. Object Request Broker.

The ORB supports many functions in order to operate consistently and effectively, but many of these functions are hidden from the user of the ORB. It is the responsibility of the ORB to provide the location transparency, or in other words, to make it appear as if the object is local to the client, while in reality it may reside in a different process or machine. Thus, the ORB provides a framework for cross-system communication between objects. This is the first technical step toward interoperability of object systems.

Another step toward object-system interoperability is the communication between objects across platforms. The ORB allows objects to hide their implementation details from clients. This includes programming languages, operating systems, host hardware, and object locations. Each of these can be thought of as a transparency, and different ORB technologies may choose to support different transparencies, thus extending the benefits of object orientation across platforms and communication channels.

There are many ways of implementing the basic ORB concept. For example, ORB functions can be compiled into clients, can be separate processes, or can be part of an operating system kernel. These basic design decisions might be fixed in a single product; or there might be a range of choices left to the ORB implementer.

There are two major ORB technologies:

- The Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA) specification

- Microsoft's Component Object Model

Remote method invocation can be implemented in a Multilanguage RMI system, such as CORBA RMI, or via a single-language RMI system, for instance, Java RMI.

C. USING METADATA TO ADDRESS PROBLEMS OF SEMANTIC INTEROPERABILITY

The challenges to achieving semantic interoperability is that semantic information can be implicit: embedded in the software. Moreover, one must be able to extract the implicit information and make it explicit as metadata. It is not easy to make semantic information explicit.

- Semantic is associated with data and procedure names, screen layout, etc.
- Semantic information is embedded in the application code and the design assumptions of data administrators and programmers

There is no universally agreed-on way of presenting semantics explicitly, so metadata specifications are potentially semantically incompatible among themselves. Even when metadata is explicitly represented, semantic interoperability still depends on arguments between the user and supplier, but at higher level in the data or metadata chain.

1. The Use of Repositories

> Repositories consist of a shared integrated database of descriptors (i.e., metadata) of an organization's information systems, including both its software and its data. Metadata describes system components at multiple levels of abstraction[10].

A repository manager manages the repository's contents. A repository provides a built-in model of metadata that may be used as the basis for documenting a system's requirements, design, implementation, and so on.

2. Tools for Specifying and Extracting Metadata

In addition to repositories, there are two other tools to extract metadata.

a. CASE (Computer-Aided Software Engineering) Tools

A CASE tool is a tool used to create structured semantic information about an information system. The user of a CASE tool can compare and relate the semantic information represented by these constructs to design semantic interoperability into the information system.

There are two general classes of tool objects:

- **Upper CASE**, which represents the business terms, organizational units, business functions and process, and data entities.
- **Lower CASE**, which represents system components such as databases, user roles, screens, etc. Lower CASE components are intended to support upper CASE objects.

---

[10] Using Metadata to Address Problems of Semantic Interoperability in Large Object Systems. S. Heiler, J. Miller, and V. Ventrone. IEEE, 1996

b. System Integration Tools

Integration tools extend the definition of CASE tools to include facilities for describing existing system components. Integration tools represent data stores, mappings, and transformations in largely symbolic terms that both reduce the labor of the developer and create semantic information that may be used to resolve semantic interoperability challenges.

CASE and system integrations can be used to assist the system analyst in detecting and resolving semantic incompatibilities. However, these two types of tools present the following challenges:

- They are not integrated with the application environment
- The content of the repository cannot be considered as active

D. CHALLENGES IN APPLYING MIDDLEWARE COMPONENTS TO AD HOC WIRELESS NETWORKS

In the previous section we described some of the requirements for realizing semantic interoperability in ad hoc networks and the various components that must be present to allow seamless access to different platforms. We also defined the role of middleware, in particular, object request brokers, in providing interoperability. However, these middleware components have been developed to enable transparent access to heterogeneous, distributed resources in wired networks, excluding or not taking into consideration wireless ones. Middleware for distributed computing needs to encompass wireless computing.

In general, mobile computers, laptops and personnel digital assistants (PDA) are equipped with more than one communication interface. A common characteristic of these interfaces is that they offer low-bandwidth or low-quality connections compared to traditional wired networks.

The networking options for a mobile host are more complex than those of a fixed host. For distributed applications designed with more static network conditions in mind (such as COBRA middleware), this environment poses a substantial challenge. The extra functionality required to deal with this environment can either take the form of mobility-enhanced applications or of special mobility support on the mobile hosts, or both.

Another problem is that the processing power and memory resources available on many mobile devices are limited in comparison to those of typical desktop machines. This restricts the user of a mobile device in that only a limited number of applications may be available. Moreover, the functionality of available applications is often limited.

A third problem, associated with mobility rather than network connectivity or hardware limitations, is how to locate mobile devices. A mobile device may be moving from one point of attachment to another. Therefore, a routing overhead is needed to determine the location of the host in the network. So, for a host to maintain an efficient routing algorithm, it must reserve and control resources. This routing overhead would limit the computation capabilities of the wireless device and constrain the addition of software components such as middleware functionalities that provide interoperability in heterogeneous networks.

E. INTEROPERABILITY SOLUTIONS IN WIRELESS NETWORKS

As stated above, there are several problems in applying middleware components to wireless networks. Such problems are related basically to the nature of wireless architecture and its limitation regarding processing, power, and networking capabilities. In this section we discuss possible solutions to realize interoperability in ad hoc wireless networks.

The following specifications can be applied in order to realize semantic interoperability in ad hoc wireless networks.

a. Wireless User Infrastructure

Wireless user infrastructure consists of several functional components, starting with a mobile device with sufficient memory, an appropriate display, and communications functionalities. Several suitable models are now available, such as the Palm Pilot, a personal digital assistant (PDA) with a wireless transmitter and receiver and antenna, with computing functions. These devices are oriented toward either communication or computing. As these devices gain more functions and grow in storage and processing capabilities, they will need an operating system to manage resources. A general-purpose operating system (OS) is not suitable for these devices because of their real-time requirements, processing power, limited memory, small screen size, and typical applications—such as voice. These devices need an OS with a small footprint, at most 1 Mbytes, and reduced storage needs. Nearly all OS vendors have attracted developers of applications for handheld and smaller devices. Since Unix has been used widely on the Internet and in other computing environments, a stripped-down version requiring a smaller footprint may become important for mobile applications.

b. Wireless and Mobile Middleware

Middleware unites different applications, tools, networks, and technologies, giving users a common interface. Mobile middleware is an enabling layer of software to connect wireless applications with different mobile networks and operating systems without introducing mobility awareness—the need to adjust to wide variations in bandwidth and resulting delays, and changes in user location—in the applications, Middleware gives applications better response times and far better reliability. Typically, middleware uses optimization techniques, such as header compression, delayed acknowledgments, and concatenation of several smaller packets into one to reduce wireless network traffic. Some middleware supports intelligent restarts, which take the user to the break point after disconnection instead of back to the beginning.

ExpressQ from Nettech (Broadbeam Corporation (formerly called Nettech Systems, Inc.), is a wireless infrastructure provider, offering an award-winning wireless development platform and SystemsGO), is a mobile-messaging middleware product that uses logical name addressing to allow network and device independence, supports several wireless networks, provides multiple application programming interfaces (APIs) for developers, and lets mobile devices run different operating systems.

c. Wireless Application Protocol (WAP)

Using a common set of applications and protocols, WAP facilitates interoperability among different wireless networks, devices, and applications. WAP uses a microbrowser as the client software and supports text, graphics, and standard Web content. WAP uses, also, a proxy gateway to translate WAP requests from mobile clients to protocols employed by the information server on the other side. Encoders translate the

content coming from the server into compact formats to reduce the size of data over the wireless network. This infrastructure lets mobile users access a wide variety of content and lets application developers, using proven and existing technologies; build applications that run on a large base of mobile terminals.

The World Wide Web Consortium (W3C) has developed several open recommendations for extending existing Internet standards so that wireless devices can fully access the Web and its information base.

The W3C has devised several recommendations to allow Web device independence, content reuse, and network-friendly encoding:

- The Extensible Markup Language (XML) for richer semantic information
- Improved cascading style sheets and Extensible Style sheet Language to further separate content from presentation
- A document object model defining a language-independent API that applications can use to access and modify HTML and XML documents' structure, content, and style

These W3C specifications, along with the WAP specifications, will enable a wide range of wireless networking applications.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CASE STUDY: A WIRELESS AD HOC NETWORK FOR THE BATTLEFIELD (WAHB)

There exists a large number of application scenarios in which wireless access to heterogeneous information sources would be of great value. Tactical communication for military operations, disaster situations, and emergency crisis management are just a few applications of ad hoc wireless networks. There is an expectation that the rapid advance in mobile devices such as laptops and Personal Digital Assistant (PDA) will enable anywhere "always on" communication to be more accessible. Such a wireless data network is expected to support communication among people who have a common purpose to form a temporary community, such as a coalition of special operations forces.

In this chapter we describe an ad hoc wireless network that is suitable for use on a battlefield. We call this network a Wireless Ad Hoc Network for the Battlefield (**WAHB**).

Some of the requirements we envision for a WAHB are efficient utilization of the network resources over radio frequencies, accommodation of a large number of nodes, and support of high volumes of traffic (e.g., to support multimedia applications).

A. WAHB ARCHITECTURE AND CONFIGURATION

Our WAHB architecture is built upon wireless domain, location-aware, and self-organizing networks. WAHB is composed of a set of mobile nodes that have the following characteristics:

- Each node in the wireless ad hoc network consists of a router integrated into a single device such as a laptop or handheld computer

- Each node is equipped with a wireless transmitter and receiver using antennas that can be omni directional (broadcast), highly directional (point-to-point), or a combination thereof

At a given point in time, depending on the nodes' position, transmitter and receiver coverage patterns, transmission power level, and co-channel interference levels, wireless connectivity in the form of a dynamic ad hoc network exists between the nodes.

The basic infrastructure technology to support mobile operation in WAHB can be defined in four levels, as shown in Figure 10.

At the lowest level, we have the basic low-power hardware and firmware to permit operation in motion. It is the untethered node. At the next level, the untethered nodes are tied together with networking technology to provide a robust (in terms of routing protocols and transmission devices) communication network. Finally, to fully exploit the mobile communications capability, a mobile computing environment is needed which can deal with changing and sporadic connectivity.

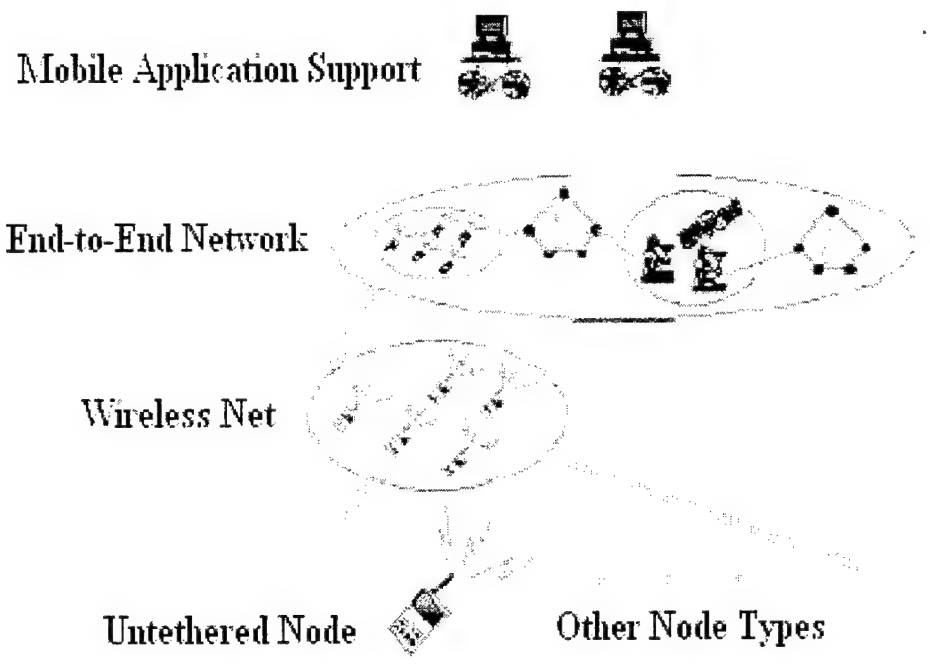

Figure 10. Basic Infrastructure Topology.

A set of WAHB nodes, as shown in Figure 11, is essentially a mobile routing infrastructure and can operate in isolation or be connected to the greater internet via extension-routing services. In this mobile infrastructure, users can change their positions as necessary. Each node contains a router. Thus, the routing infrastructure can move along with the end device. In addition, the infrastructure's routing topology can change, as can the addressing within the topology. In this paradigm, an end user associated with a mobile router (i.e., its point of attachment) determines its location in the wireless network.

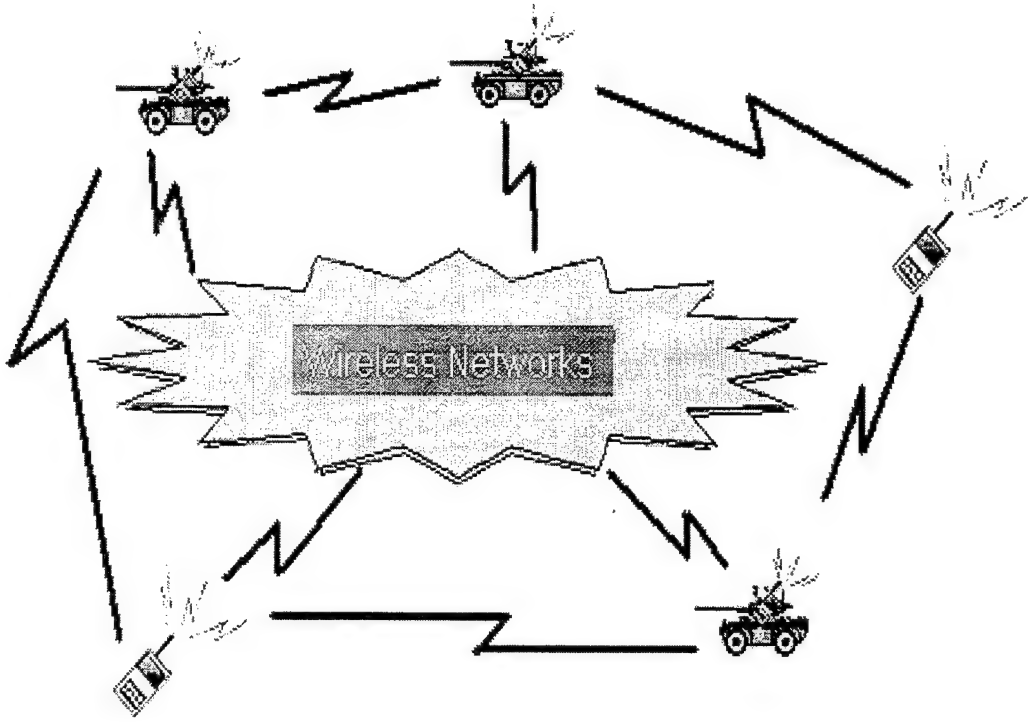

Figure 11. Wireless nodes in WAHB.

B. WAHB CHARACTERISTICS

A system based on WAHB is intended to operate under widely varying environmental conditions. It is envisioned to be relatively large, dynamic, and heterogeneous, with hundreds of nodes per mobile domain. WAHB can also be applied with a small number of nodes, depending on the nature of the mission.

WAHB has several characteristics that differentiate it from fixed multi-hop networks, such as the following:

- Dynamic topology: because nodes can move arbitrarily, the network topology can change randomly and rapidly. This gives more flexibility to end users to freely move and have access to information systems to

accomplish the mission at hand. However, adjusting transmission and reception parameters, such as power, can impact the network topology.

- Untethered nodes: they consist of laptops or hand-held communication devices. These untethered nodes are expected to support as much as possible the required bandwidth, range, networking, mobility, and constraints on power usage. Note that wireless does not necessarily imply mobile. For example, consider fixed satellite and ground stations communicating with each other to establish a link. This is not a mobile system. Similarly, mobile does not necessarily imply wireless.

Some of the drawbacks of wireless networks in general are the following:

- Bandwidth constraint**:** wireless links at present have significantly lower capacity than their hardwired counterparts. Hence, congestion is more problematic in wireless links than in wire-bound networks.

- Energy constraint operation**:** some or all of the nodes in WAHB may rely on batteries for energy. For these nodes, power conservation is very critical. This places constraints on the area of usage and the duration of time for which a mobile node can be powered on. Therefore, power replacement strategy and policy must be defined for each type of mission and node.

- Wireless vulnerabilities and limited physical security: mobile wireless networks pose some challenges related to protecting the devices and the information exchanged among them.

C. WAHB REQUIREMENTS

The most forward deployed tactical military units are heavily dependent on wireless communications, and in some circumstances the networked computing operations available in the wired world do not work, or do not work well, in the wireless world. Wireless communications are often unreliable, with fading, have high bit error rates, and sporadic connectivity. They have relatively short ranges, high latency in transmission, and have low data rate capacity when compared to wired network services.

The requirements for a WAHB system are to provide real-time information collection and management for tactical data-sensor fusion and decision-making, robust wireless communication with voice, data, video/graphics, and geo-location capabilities. The system is to be flexible and rapidly deployable. Also it should be adaptable to many different environments, such as in cities and rugged rural areas.

Moreover, one of the primary requirements of WAHB is to digitize the battlefield. The lower echelon commander's command and control today is based on the use of radio communication. In digitizing the battlefield, the military is seeking to harness the power of the computer to help the commander and his forces better understand their situations, improve force synchronization, and enhance combat effectiveness. For example the Global Positioning System can be used provide troops with their locations. This information can automatically be reported to other friendly forces over the wireless network.

WAHB, as shown in Figure 12, is envisioned to provide connectivity and access to services for wireless mobile users whether the end user belongs to a brigade, battalion, or company. WAHB is also envisioned to provide seamless and transparent

communication among these categories of groups and also with any other friendly groups on the battlefield.

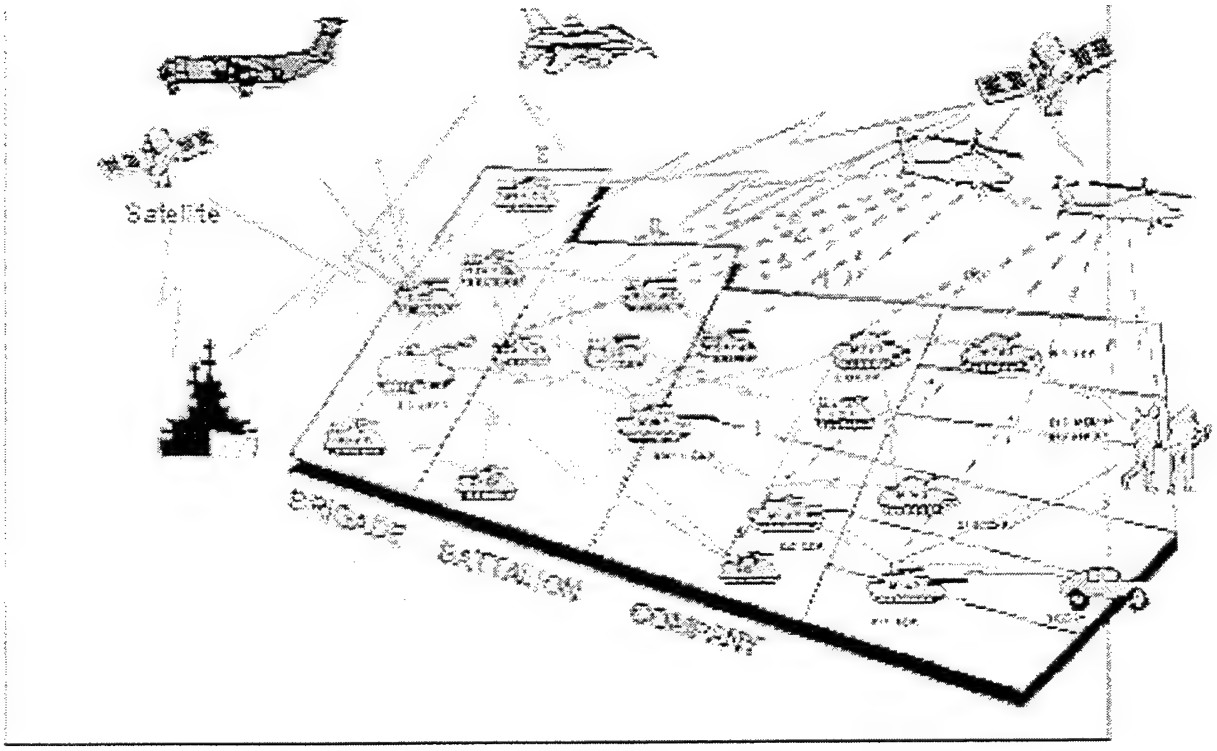

Figure 12. Digitizing the Battlefield.

D. SCENARIO

As stated before, the motivation for our research is to digitize the battlefield or any other area that necessitates military interventions via the use of wireless ad hoc networks. To achieve this goal, military teams coordinate and share different kinds of information which are stored on different nodes of the network. In this case, access to information must be driven down to the lowest possible tactical level. Communications

systems do not have to follow the chain of command, and an end user must be able to move and access information in the ad hoc network in a seamless way.

We illustrate the above requirements via a scenario consisting of a rescue operation in a disaster area. The environment size is 2000 by 900 meters. This size is scaled according to the range of transmitters. This scenario represents some sort of disaster area in a region that lacks an operable (e.g., may have been destroyed in a fire or flood) telecommunication infrastructure.

This mission has been granted to a special military team along with a special firefighter team. These two teams must coordinate their rescue-operation efforts by way of sharing information with one another.

Every rescue team member could have a personal communication device with wireless ad hoc network capabilities. These personal communicator devices (laptops, web-enabled cell phones) are capable of communicating with each other and with relay nodes that are mounted on vehicles, such as helicopters, or ground vehicles, as shown in Figure 13.

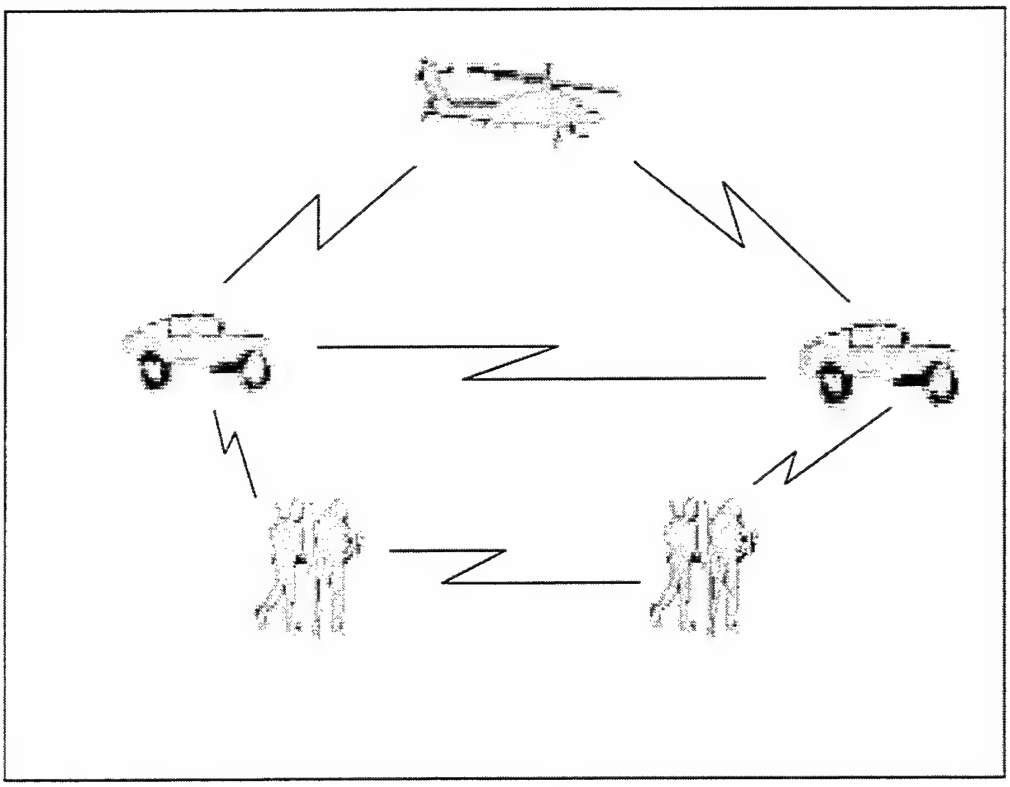

Figure 13. First View of the Scenario Area.

The scenario is characterized by:

- Slow nodes and some very fast nodes (those mounted on a car or a helicopter).
- Node movement, where approximately 95 percent of the nodes are moving slightly while the remaining 5 percent are changing their position very often.
- Traffic is spread all over the network, but necessarily uniformly distributed.
- There is low interference from the other nodes in the ad hoc network area

We mentioned before that the military special team is going to coordinate the operation with a firefighter special team. We assume that both teams are using the technology of ad hoc wireless networks.

In this scenario, both teams may need different kinds of information about the area that should be evacuated, such as the weather conditions, the geographical characteristics, and the distances to nearest hospitals. For the mission to succeed, there is a need to be able to obtain relevant information without having to specify the source. The system should be able, whenever a query is asked, to automatically find the best matching information source, both in terms of content and current connection properties.

One of the most challenging aspects of this scenario is the need to provide reliable access to distributed information sources in hostile environmental conditions. Power outages and disrupted telephone lines are two of the most common consequences of natural disasters. Thus, it is not possible to rely on traditional computer networks to obtain access to information sources. Additionally, various kinds of information will either be needed or provided by rescue personal working in the field, possibly in areas where, even under the most favorable conditions, access to a computer network would be difficult, if not impossible. A solution to this dilemma, as stated before, is to rely on wireless communication links to provide access to the information sources. However, in addition to providing wireless connectivity and interoperability, one must implement an efficient routing protocol strategy, and use mediators to get seamless access to information on computers belonging to other teams working in the joint force rescue operation.

In our scenario, each member in the rescue team needs to be able to send messages to the other members of the team according to the operation's rules. These rules state that any member of the team might send/receive a message to/from a higher echelon's device, if necessary.

In our scenario, we are going to have, as stated before, two types of nodes: fast-moving nodes (generally mounted on ground vehicles), and slow-moving nodes held by the team members. Based on node repartition and mobility patterns, our model of the network is based on hybrid protocol combining table-driven and on-demand routing techniques. The fast-moving nodes use on-demand routing, whereas low mobility nodes use table-driven routing protocols. This hybrid approach will prevent the frequent update of routing tables in slow-moving nodes.

The candidate terminals for routers are only slowly moving terminals. Normally, when a source node does not find a destination node in the cluster where it belongs, the routing-path discovery procedure is started. However, in our approach, the paging signals to the nodes in the cluster are received by only low-mobility terminals. When the low-mobility terminal receives the paging signals it uses the table-driven approach. The updating rates of routing tables between low-mobility terminals are comparatively low. Hence, the prepared routing table can be used without paging to find the routing path.

We now apply this routing approach scheme to our scenario, because as we mentioned before, there are two types of mobile nodes, those with low mobility and those with high mobility.

Assume each rescue team consists of eleven members. Regardless of the appurtenance of the team to the military or the firefighter team, we assume that both

groups are going to cooperate in the operation. Each eleven-member team is led by a team commander who has a highly capable device such as a laptop with more processing, memory and storage capabilities than team member's devices, and a main node (clusterhead), which is going to act as a router. Another member of the team has to have a device that is capable of taking over the routing operation in case of main-node failure. This procedure is taken into account to provide a rudimentary level of fault tolerance. The team topology is flexible because in case of the need to split the eleven-member team, we have always two devices that can play the role of network router. The high-mobility nodes that are carried by ground vehicles consist of powerful wireless devices that can host multiple types of information systems. These nodes, as mentioned before, use the on-demand routing approach.

Our model supports the hierarchical command structure of the military. In fact, our scenario can be extended to bigger operations, shown in Figure 14, as long as each cluster head is provided with enough computational resources.

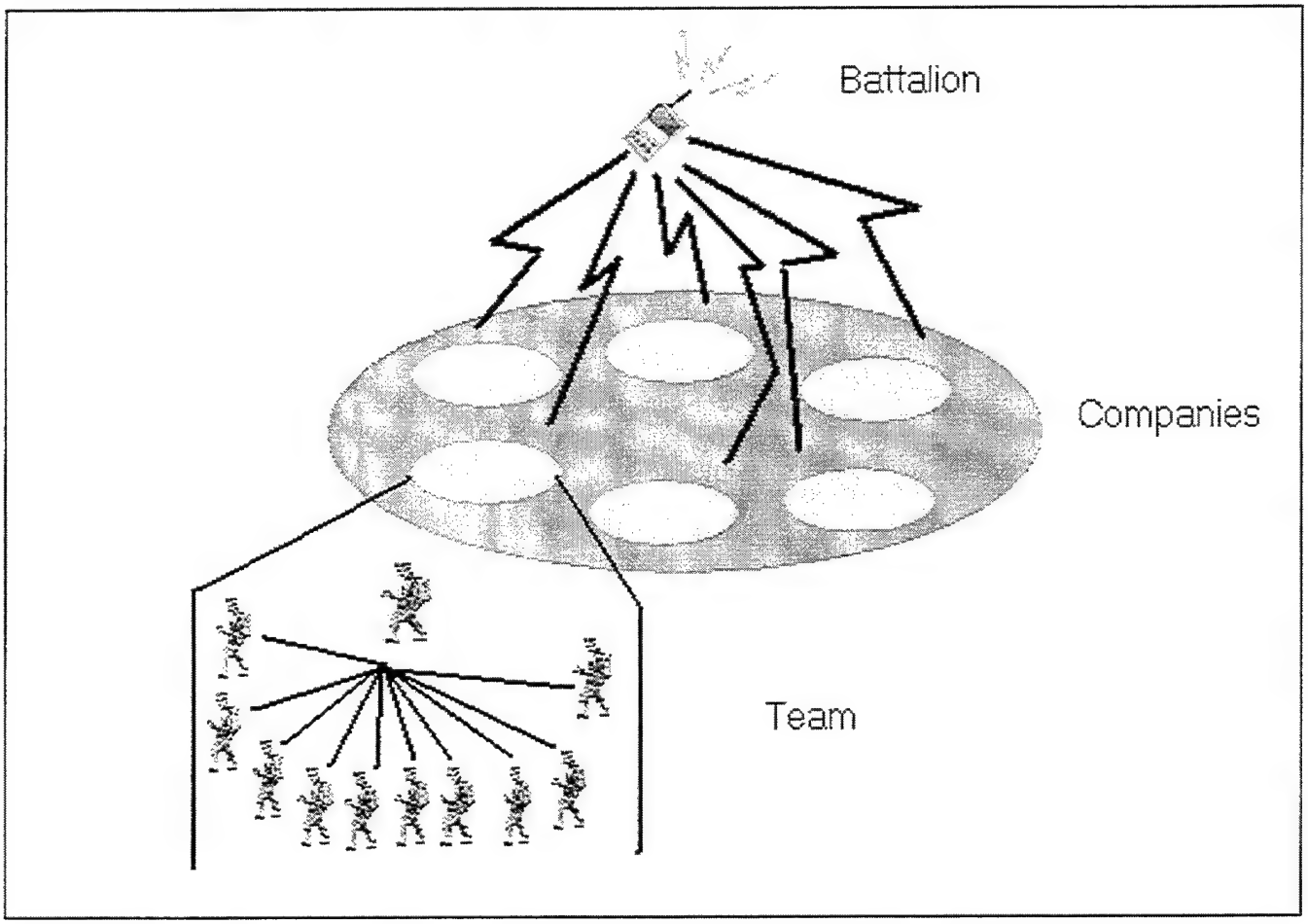

Figure 14. Clustered Military Levels.

Next we address the aspects of our model that support interoperability between the joint rescue team. Our model incorporates mediators, which are middleware components that homogenize, integrate, or otherwise processes information to make it available to high-level applications and services.

Our system is based on heterogeneous, distributed, autonomous information sources. In order to offer our rescue team transparent access to these sources, semantic heterogeneities must be overcome. This can be provided by homogenization of the mediators that transfer the information to an ontology shared by all components in the system. The use of mediators can work well in fixed networks because constraints on storage, processing, and networking are not as severe, in general, as those for wireless mobile networks.

In order to enable mobile users to access information from heterogeneous, distributed sources, we should make the data readily available at the mediators. The mediators need to be able to materialize at least portions of the data they offer. These mediators, to run efficiently on mobile users, must be light and do not request a lot of machine resources. In our model, the mediators rely heavily on the usage of standardized middleware that make it easy for both parties in the joint operation to communicate with each other.

In addition, the system is going to provide mobile users with a query interface that develops and controls a query execution plan. Given a query, the query interface has to determine appropriate mediators to answer it. In addition, this query interface has to control query execution relying on appropriate strategies to handle failures, which are likely to occur more frequently in wireless networks than in traditional wired networks.

Figure 15 shows a layered model containing the different components of a wireless device that would be able to communicate with another node in the network in a seamless way.

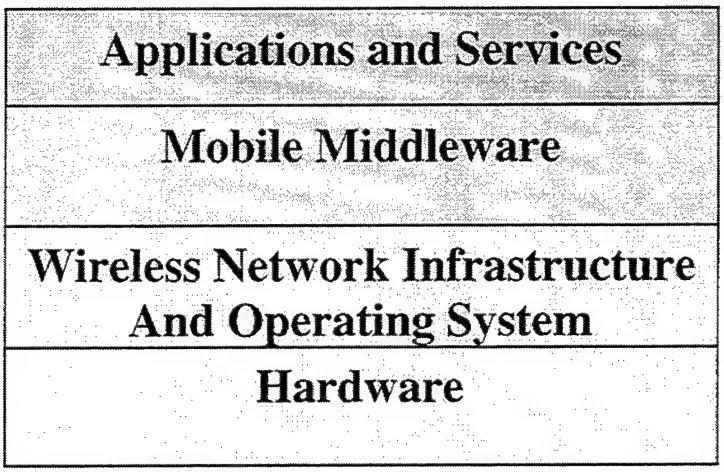

Figure 15. Wireless Layered Model.

The mobile middleware represented in Figure 15 consists of an enabling layer of software that provides wireless users in the ad hoc network with the ability to communicate in a seamless way. Its role is to unite different applications, tools, networks, and technologies, giving users a common interface. This middleware layer can also reduce the content size and format to better adapt to the characteristics of specific wireless networks and the limitations of the mobile devices, possibility resulting in the system meeting the quality-of-service requirements that the users have espoused.

The wireless network infrastructure in Figure 15 consists of:

- The transport layer (TCP/UDP)
- The Internet protocol (IP) along with the routing protocol (table-driven routing protocol for low-mobility nodes, and on-demand routing protocol for high-mobility nodes)
- The medium access control

By applying the above features concerning the use of hybrid routing protocols, the use of standardized middleware, and the use of query interface to our ad hoc wireless network, the rescue team will be able to obtain relevant information without having to specify the source. Whenever a query is asked, the system should be able to find the best matching information source both in terms of content and current connection properties.

In our scenario the rescue teams are spread out in the disaster area helping people get out of collapsed buildings. For the need of one specific rescue operation, team A finds itself in need of an ambulance and a fire engine to both extinguish the fire and aid in rescuing people trapped inside the building. In order to satisfy its needs, team A needs to know the nearest ambulance and firefighter track that can fulfill the request. For this purpose, a member of team A broadcasts a query to find out the nearest available vehicles. The results of the query should return the position of each vehicle as well as its distance from the rescue operation. Upon receiving the responses to the query, the rescue team member sends a message to the desired vehicles asking for help.

Now lets see how this query is handled by the wireless ad hoc network of the joint rescue teams. Team A, using a mobile device such as a laptop or a palm pilot with a standard query interface, writes and submits the query.

First, the system starts searching for results in team A's cluster which is using table-driven routing protocols. If no results are returned from the same cluster, the request hits the clusterhead in order to get routed out of team A's cluster to the rest of the network. Team A's clusterhead is going to send the query to all the hops (i.e., clusterheads) next to it in the wireless ad hoc network, as shown in Figure 16. Note that the routing protocol used among clusterheads is the on-demand routing scheme. For their

turn, the destination clusterheads forward the request to their cluster nodes and to the next clusterhead. At the end of the query operation, whichever nodes satisfy the request return the results to the requesting node. The resulting list provides team A with information about the different vehicles availability, distance to team A, and contact procedures. Upon receiving this list, team A chooses the most appropriate vehicle depending on availability and distance, and contacts that vehicle for assistance.

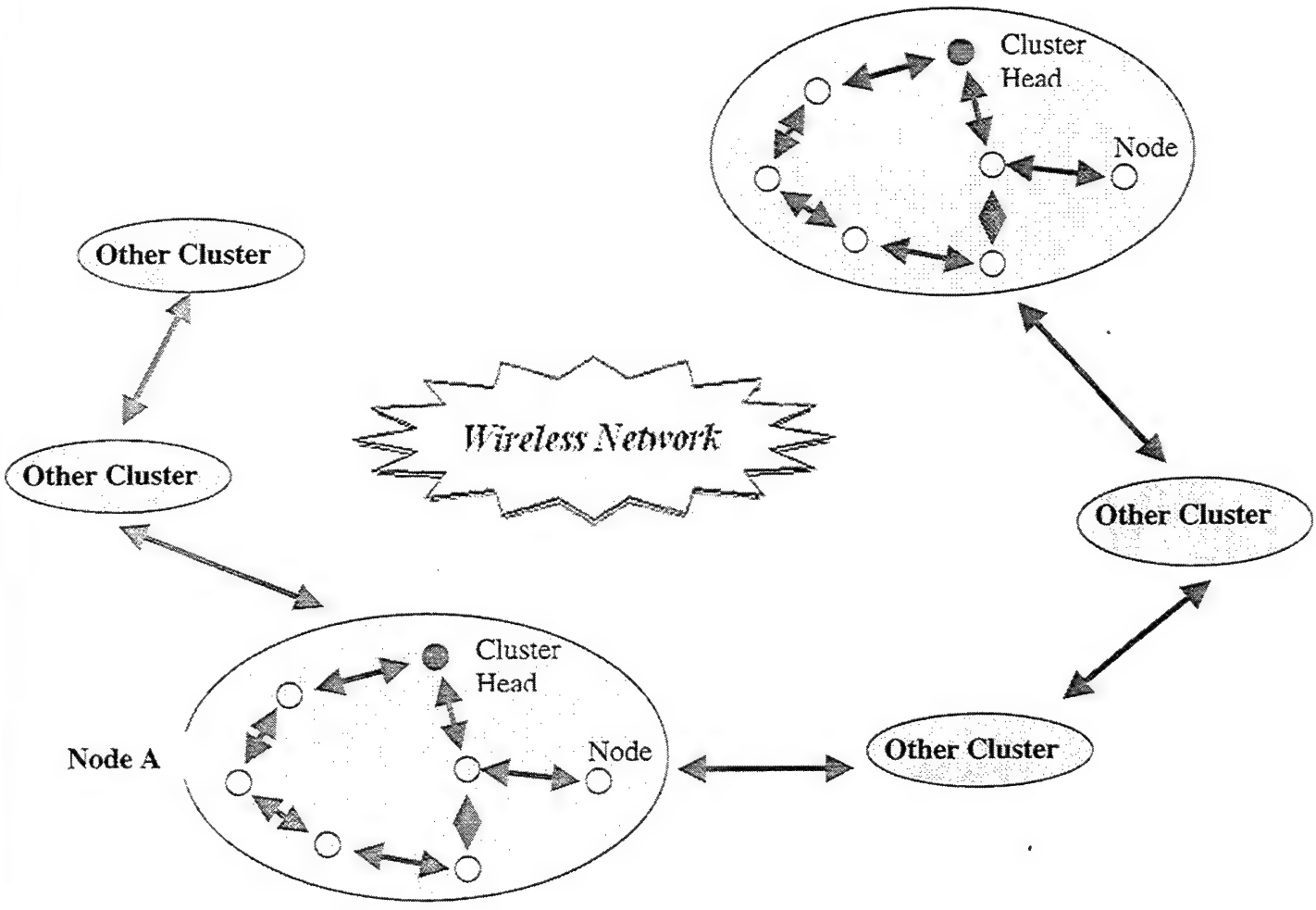

Figure 16. Communication Among Clusters in the Rescue Operation.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION

In this thesis we have explored the role that semantic interoperability plays in the realization of information systems that are comprised of nodes making up ad hoc wireless networks.

We began our research into semantic interoperability by first defining the terms 'ad hoc network' and 'semantic interoperability.' Much of the previous work on semantic interoperability had been done in the context of distributed and multidatabase systems, while the focus on interoperability in the network community has been on building bridges between communication protocols.

We used the prior work as a foundation upon which to define general semantic interoperability requirements for ad hoc wireless networks. We also had to characterize the nature of such networks in order to provide as basis from which to identify requirements for semantic interoperability that are specific to ad hoc wireless networks.

We identified routing protocols, wireless links, application and services, and design issues. We identified many characteristics that distinguish wireless networks from the wired ones. These characteristics can be summarized as follows: limited bandwidth, energy constraints, limited processing power, limited memory, and security vulnerabilities.

We explored the following aspects of semantic interoperability argument describers, conversion functions, planners, and the request object broker. We discovered that these interoperability functionalities can work well in wired networks, but cannot be readily applied to ad hoc wireless networks, due to the aforementioned characteristics of

ad hoc wireless networks. In order to address these characteristics, we proposed the following requirements for the network:

- Adequate wireless user interface consisting of:
  - Mobile device with sufficient memory
  - Operating system with small footprint (300 KB to 1 MB) that can handle real-time requirements, processing power, small screen size, and typical wireless applications
  - Transmitters and receivers that can handle the desired transmission range and bandwidth
- Mobile middleware that is
  - Able to connect application with different mobile networks and operating systems
  - Can reduce the content size and format to better adapt to the limitations of mobile devices and allow better response times
  - Able to hide the differences in applications from a mobile user.
- Wireless application protocols that
  - Are able to translate requests between mobile users
  - Use a micro browser as the client software
  - Use an extensible markup language (XML) for rich semantic information

We illustrated our set of requirements for semantic interoperability via a case study consisting of a joint forces rescue operation.

Wireless ad hoc networks are an emerging technology that presents several challenges. A lot of research is being conducted to address issues related to routing protocols, medium access control, transport protocol, security, and interoperability. Semantic interoperability, here, is only one piece of the puzzle. It needs to be addressed in the context of the aforementioned issues. We have done so, with particular emphasis on ad hoc wireless networks that are made up of heterogeneous sub networks.

The requirements we have outlined for semantic interoperability can be applied to ad hoc wireless networks to be used by the military, such as by special operations forces, and joint forces that cannot rely on the use or availability of a fixed communication infrastructure.

As stated above, there are several problems related to wireless ad hoc networks and interoperability that can be addressed:

- **Performance of routing protocols**: in the design of routing algorithms, maximizing the network's capacity to carry user data and minimizing the end-to-end packet delay are the key considerations. The decisions made in the design of routing protocols are also dependent on the underlying link layer and physical layer technologies. Therefore, in order to build or simulate an efficient routing protocol, you have to consider the issues stated above.
- **Study how much information can be lost in mediation operations**: mediation activities involve the operation of determining the expectations that two applications have for a value "v". During this operation, mediators must describe the value's arguments in order to execute the

right conversion. However, depending on the application, the information describing the value's argument, sometimes, cannot be totally extracted, thus resulting in an information loss. We have to decide in this case, whether this information loss affects the conversion results or not. If yes, the system must reject this query.

- **Quality of service in ad hoc wireless networks**: in a mobile environment, providing strict quality of service guarantees, and robust service are competing requirements. The ability of a network to provide a specified quality of service depends on the performance of the links and nodes within the network, the traffic load, and the adaptive control algorithms operating at the different layers. A comprehensive approach to supporting QoS in ad hoc networks involves the use of adaptive control algorithms at the different layers. Adaptation begins at the lowest layers and moves upwards only when the lower layers can no longer maintain the QoS at the desired level. By localizing the effects of the changes within the network and within the layers, this approach limits the quantity of network resources needed to maintain QoS. Degradation in QoS that is obvious to the higher layers as a result of unfavorable changes in the environment can be minimized by fast adaptation at the lower layers.
- **Security**: is a very important issue that must be considered. Since ad hoc wireless networks are formed without centralized control, security must be handled in a distributed fashion. This will probably mean that IP security

authentication headers will be deployed, as well as the necessary key management to distribute keys to the members of the ad hoc network.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Agrawal, P., Famolari, D.; "*Mobile Computing in Next Generation Wireless Networks*," Proceedings of the 3rd international workshop on discrete algorithms and methods for mobile computing and communications. August 20, 1999, Seattle, WA.

2. Murthy, U.; Bukhres, O.; Winn, W.; Vanderdez, E.; "*Firewalls For Security in Wireless Networks*," Proceedings of the Thirty-First Hawaii International Conference on System Sciences (HICSS'98). IEEE, 1998, pp. 672 –680.

3. Garcia-Luna-Aceves, J.J., Chaine, F.L., Ewerton, M., Beyer, D., Frivold, T.; "*Wireless Internet Gateways (WINGS)*," Proceedings of the IEEE Military Communications Conference (MILCOM), Monterey, CA, November 1997, pp. 1271-1276.

4. Corson, M.S., Macker, J.P., Cirincione G.H.; "*Internet-Based Mobile Ad Hoc Networking*," IEEE internet computing, Volume 3, July-August 1999, pp. 63-70.

5. Alagar, S., Venkatesan, S., Cleveland, J.R.; "*Reliable Broadcast in Mobile Wireless Networks*," Proceedings of Military Communications Conference (MILCOM), San Diego, November 1995, pp. 236-240.

6. Torrieri, D.; "*Future Army Multiple-Access Communications*," Proceedings MILCOM, 97, Volume 2. IEEE, 1997, pp. 650 -654.

7. Bittel, R., Loso, F, Caples, E., Young, C.D.; "*Soldier Phone: An Innovative Approach to Wireless Multimedia Communications*," Proceedings MILCOM 98, Volume 3, IEEE, 1998, pp. 903-907.

8. Stehle, R.H., Lewis, M.G.; "*Wireless Networks of Opportunity in Support of Secure Field Operations*," Proceedings MILCOM, 97, Volume 2, IEEE, 1997, pp. 676-681.

9. Tendre, M.L., Macauley D., Sudnikovich M.; "*Routing and Internetworking Within The Tactical Internet*," Conference Record MILCOM, 95, IEEE, 1995, Volume 3, pp. 921-924.

10. Josh, B., Maltz, A., Johnson, B., Chun, H., Jorjeta, J.; "*A performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols,*" Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98), October 25–30, 1998, Dallas, Texas.

11. Atsushi, I.; "*Scalable Routing Strategies for Ad Hoc Wireless Networks*," communication area, IEEE journal, 1999. Volume 17, pp. 1369-1370.

12. Ashish, A., Sulabh, A., Jatinder, P. S., Rajeev, S.; "*Performance of TCP Over Different Routing Protocols in Mobile Ad Hoc Networks*," IBM India Research Labs. IEEE, 2000, pp. 2315-2319.

13. Sandra, H., Renee J.M., Vincent, V.; "*Using Metadata to Address Problems of Semantic Interoperability in Large Object Systems*," IEEE journal, 1996.

14. Rosenthal, A., Sciore, E.; "*Description, Conversion, and Planning for Semantic Interoperability*," Conference on Data Semantics. MITRE Corporation Belford, MA, and Boston College, Chestnut Hill, MA, 1995.

15. Janis, R. P.; "*Distributed System Interoperability Perspectives*," conference on system interoperability. MITRE Corporation Bedford, MA, 1996.

16. Niki, P., Kia, M., Brigitta, K.R.; "*A Middleware-Based Architecture to Support Transparent Data Access by Mobile Users in Heterogeneous Environments*," Proceedings of the 10th International Workshop on Research Issues in Data Engineering. IEEE, 1998.

17. Mads, H., Raymond, C., Vinny, C.; "*Supporting CORBA Applications in a Mobile Environment*," Proceedings of the fifth annual ACM/IEEE international

conference on Mobile computing and networking, August 15 - 19, 1999, Seattle, WA.

18. Hung, P., Hsian, W., Ming, L.; "*Dynamic QoS Allocation for Multimedia Ad Hoc Wireless Networks*," Proceedings Computer Communications and Networks. IEEE, 1999, pp. 480–485.

19. James, N. M., Nicholas, J. C.; "*Broadband Wireless Communication Via Stratosphere HALO™ Aircraft*," Military Communications Conference, 1998. MILCOM 98. Proceedings. IEEE 1998, Volume 1, pp. 45 -49.

20. Blair, G. S., Anderson, A., Blair, L., Coulson, G., Sanchez, D.; "*Supporting Dynamic QoS Management Functions in a Reflective Middleware Platform*," Proceeding. IEEE 2000, Volume 147, pp. 13-21.

21. Alistair, M.; "Mobile Middleware for the Reconfigurable Software Radio," IEEE Communication Magazine. Volu38, August 2000, pp. 152-161.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road, Suite 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, CA 93943-5101

3. Dean Dan Boger, Code IW 1
   Naval Postgraduate School
   Monterey, CA 93943-5118

4. Professor James Bret Michael, Code CS/Mj 1
   Naval Post Graduate School
   Monterey, CA 93943-5118

5. Professor John Osmundson, C3 Academic Group 1
   Naval Post Graduate School
   Monterey, CA 93943-5118

6. Captain Raouf Hafsia, Code 32 2
   10 Rue Ibn Hamdiss El Menzah 1004
   Tunis, Tunisia